NAME: VOMMIDAPU CHINNI **BATCH NO:** 120

MAIL.ID: vommidapuchinni@gmail.com **DATE:** 19/03/24

TRAINER: Mr. Madhukar **MODULE:** Terraform

ASSIGNMENT – 3

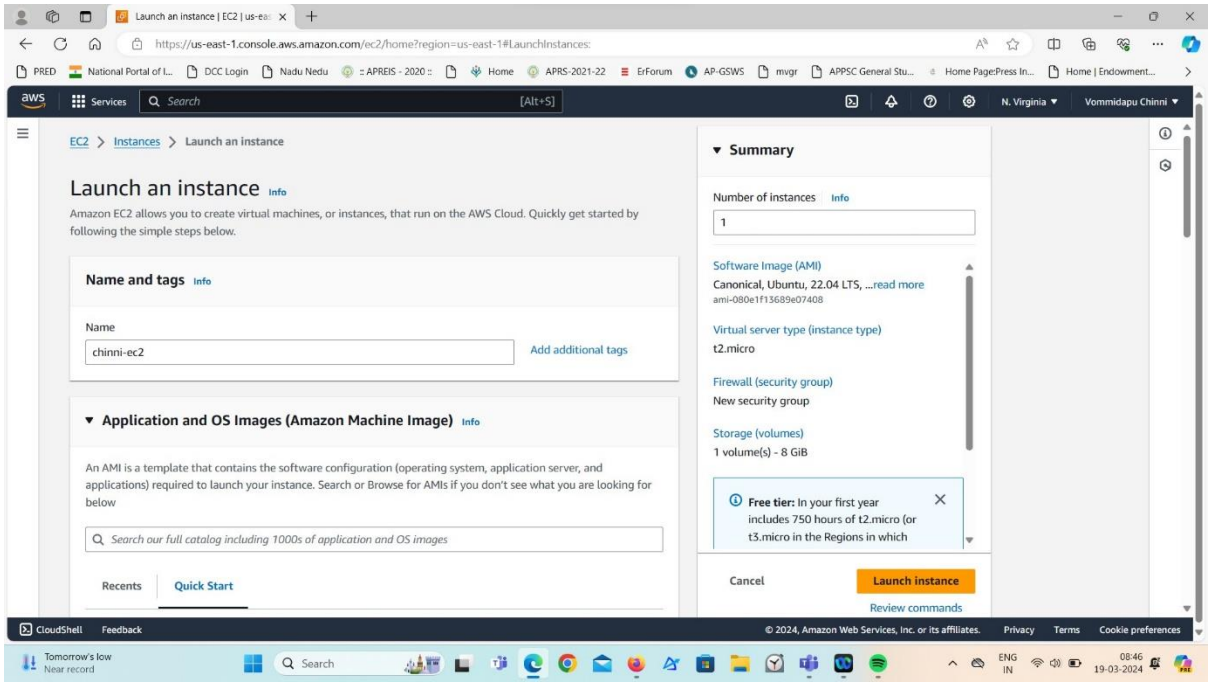
1. By using terraform create an s3 bucket and upload the file to the s3 bucket?

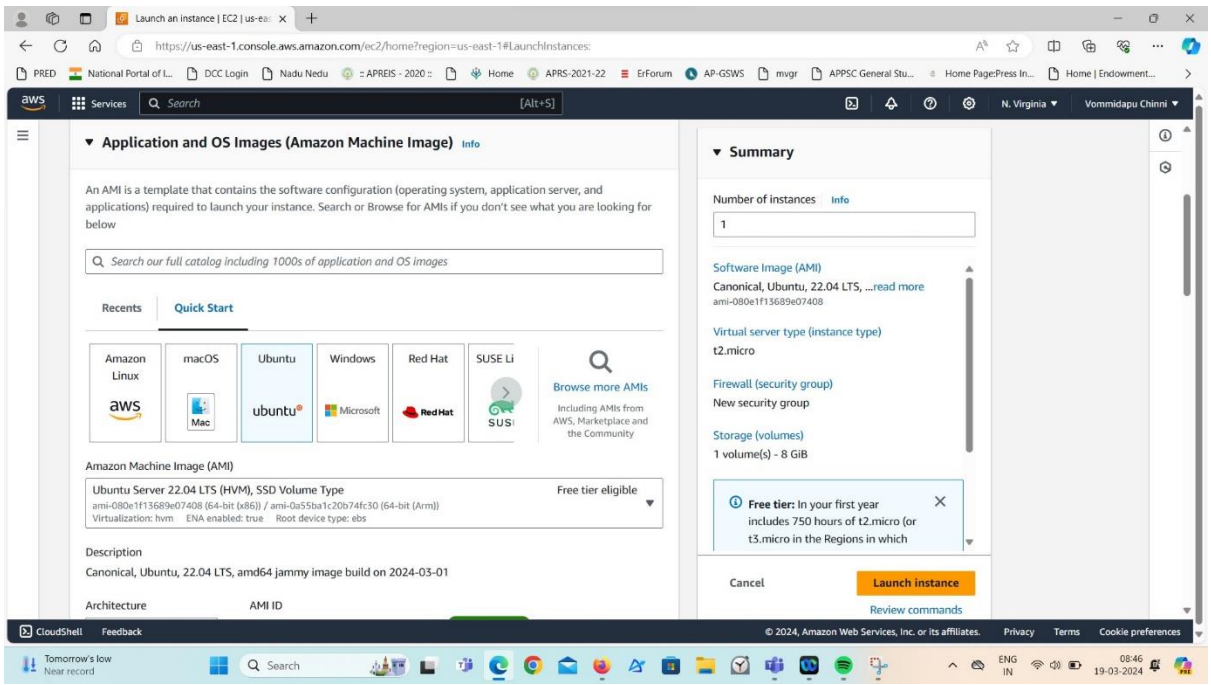
To create an S3 bucket and upload files to it using Terraform, you can follow these steps:

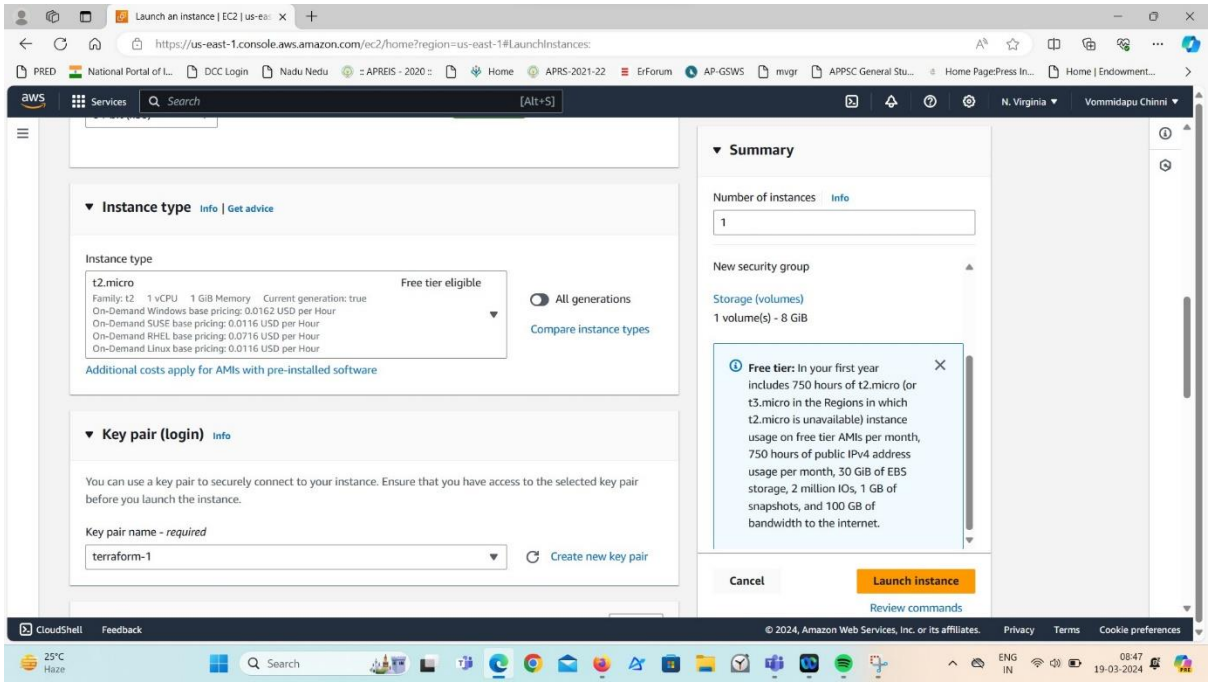
1. Launch an EC2 instance.
2. Create a user in IAM.
3. Connect to the server, Install awscli in server.
4. Install terraform in server.
5. Create terraform directory, Create terraformblock.tf, Create provider.tf, Create resource.tf.

Step1: Launch an EC2 instance

a. Click on launch instance.
b. Give name/tag.
c. Select AMI as ubuntu.
d. Instance type t2.micro.
e. Create key pair as terraform-1.

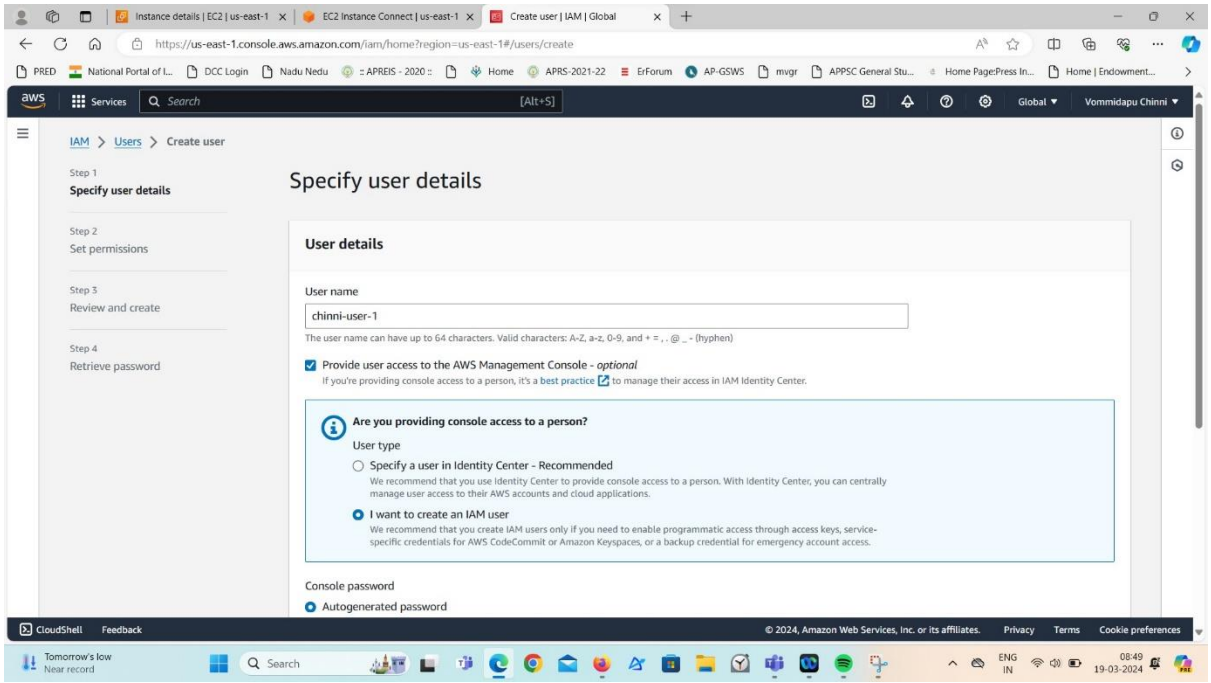

Launch an instance | EC2 | us-ea
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Services
Search
[Alt+S]
N. Virginia
Vommidapu Chinni
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
chinni-ec2
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-080e1f13689e07408
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Tomorrow's low
Near record
Search
ENG
IN
08:46
19-03-2024

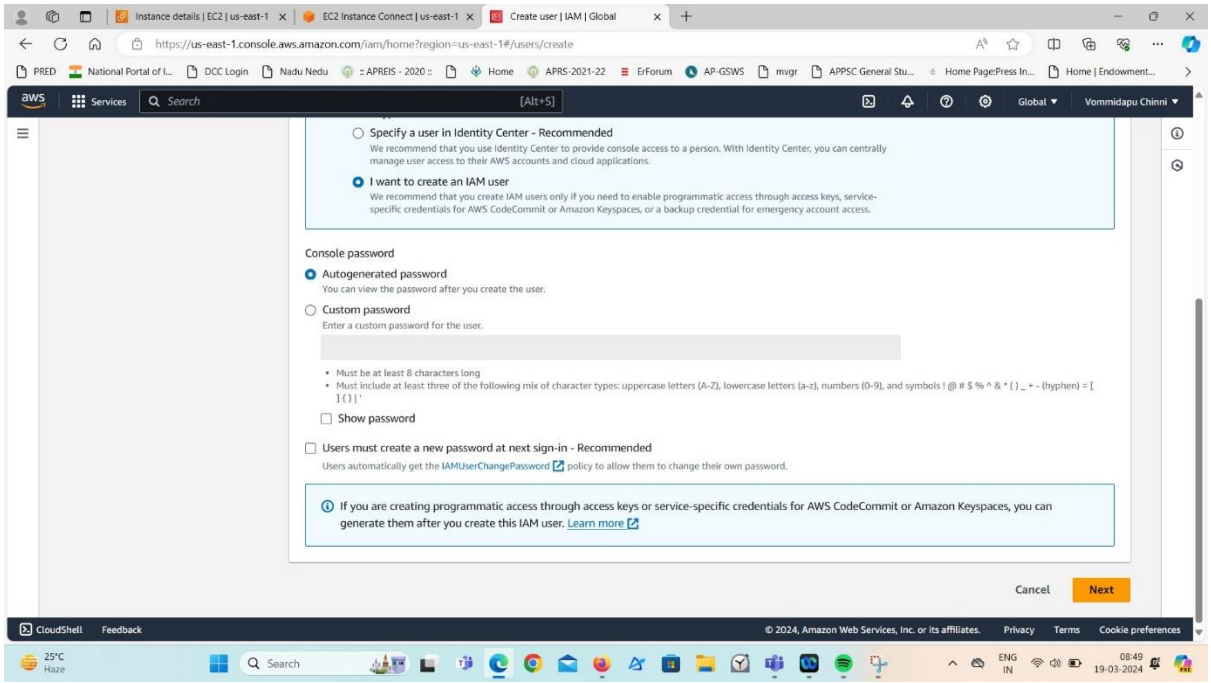

Launch an instance | EC2 | us-ea
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Services
Search
[Alt+S]
N. Virginia
Vommidapu Chinni
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-080e1f13689e07408 (64-bit (x86)) / ami-0a55ba1c20b74fc30 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Description
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-03-01
Architecture
AMI ID
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-080e1f13689e07408
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Tomorrow's low
Near record
Search
ENG
IN
08:46
19-03-2024

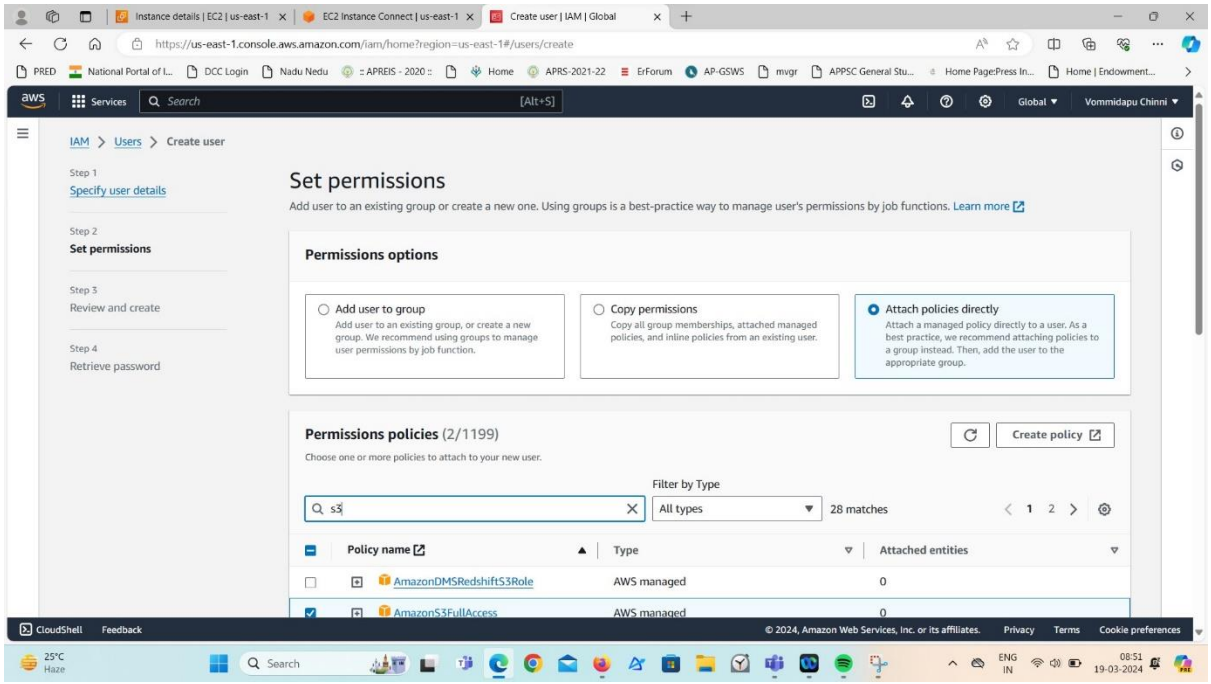


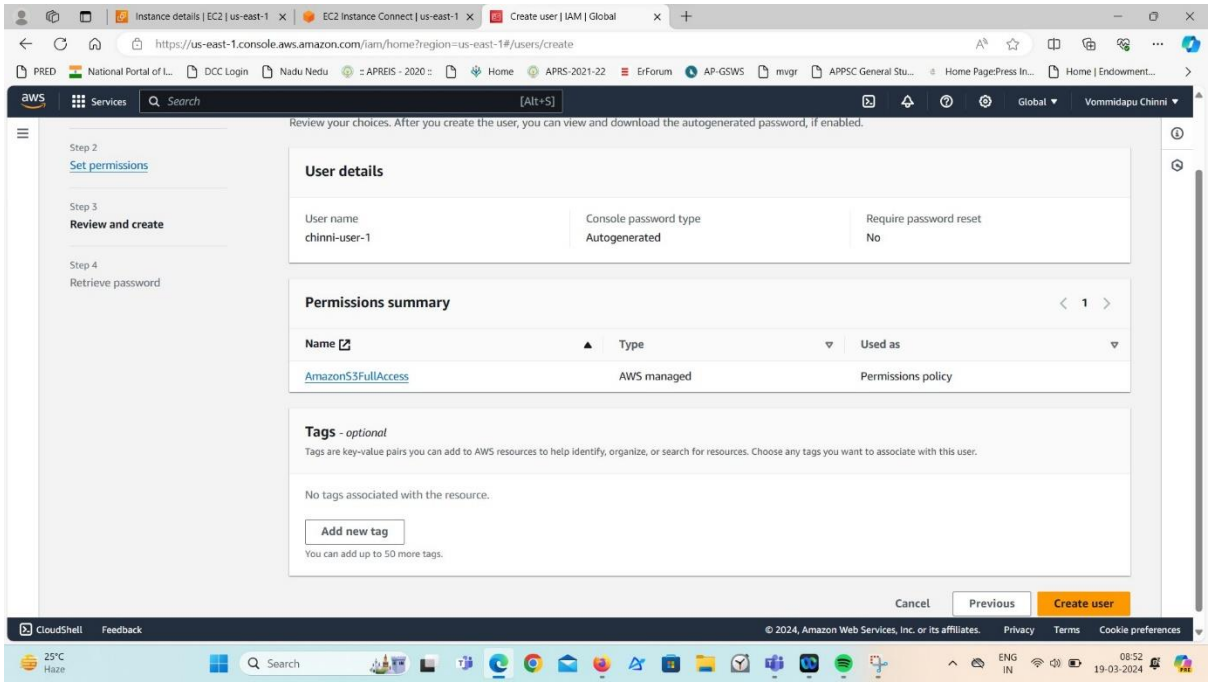
Step2: Create user IAM.

a. Search IAM, open it.
b. In dashboard click on user.
c. Click on create user.
d. Give user name.
e. Click on Attach policies directly, give s3 full access.
f. Create user.



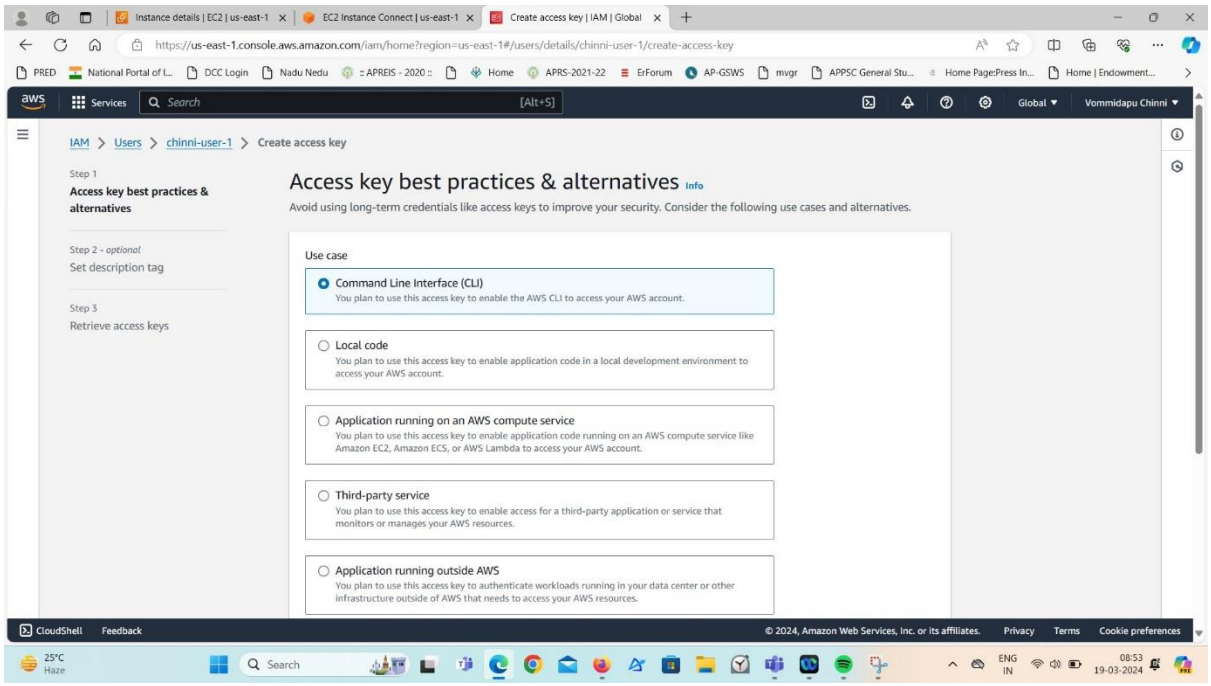

Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.
Console password
Autogenerated password
You can view the password after you create the user.
Custom password
Enter a custom password for the user.
Must be at least 8 characters long
Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '
Show password
Users must create a new password at next sign-in - Recommended
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more
Cancel
Next
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

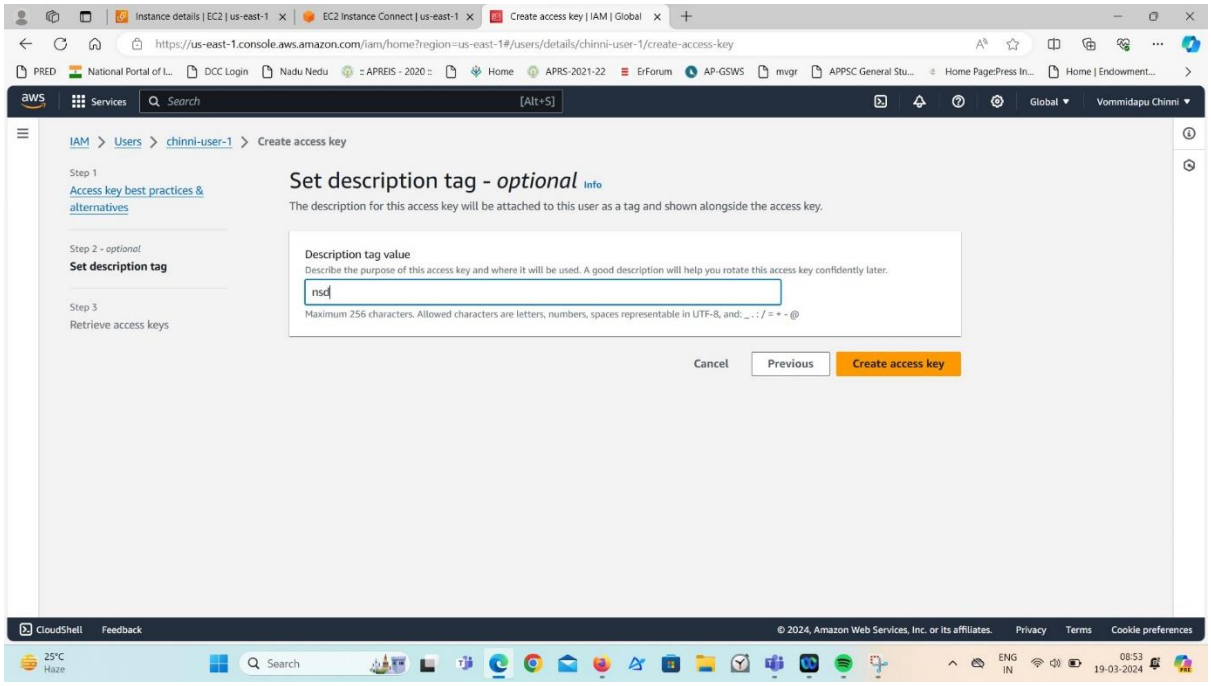

IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (2/1199)
Choose one or more policies to attach to your new user.
Create policy
Filter by Type
s3
All types
28 matches
Policy name
Type
Attached entities
AmazonDMSRedshiftS3Role
AWS managed
0
AmazonS3FullAccess
AWS managed
0
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

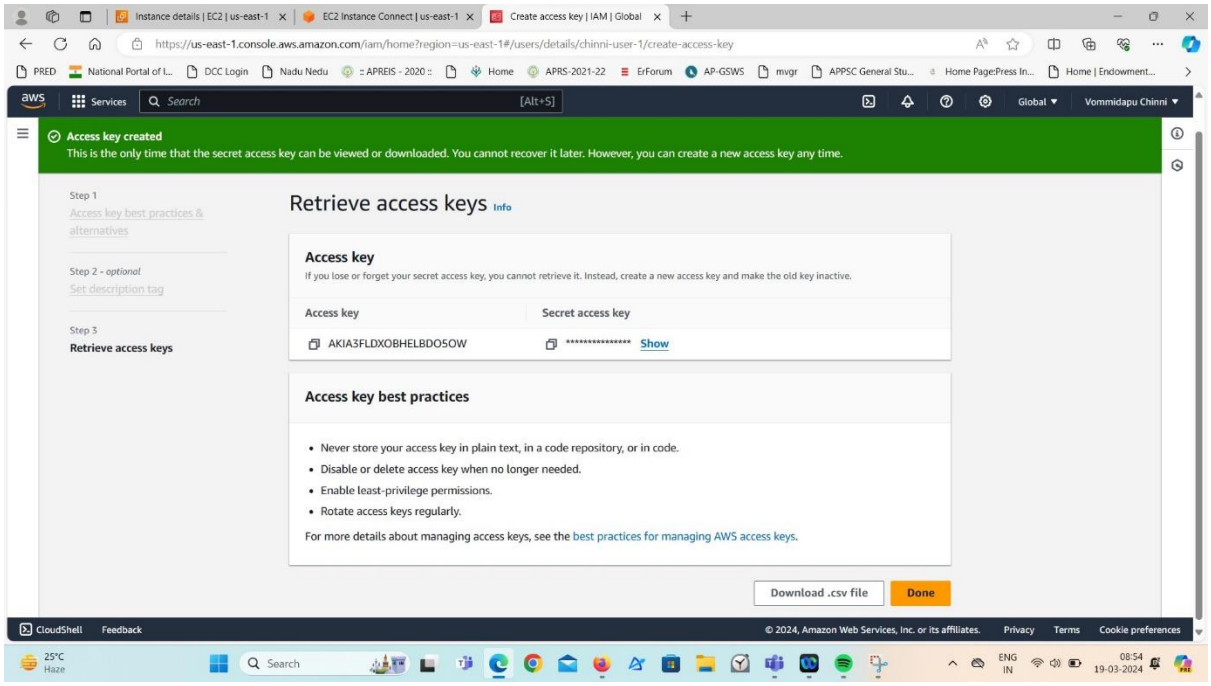


- Create an access key:

a. Click on user.
b. On right side we can see create access key, click on it.
c. Select cli, give a tag and create it.



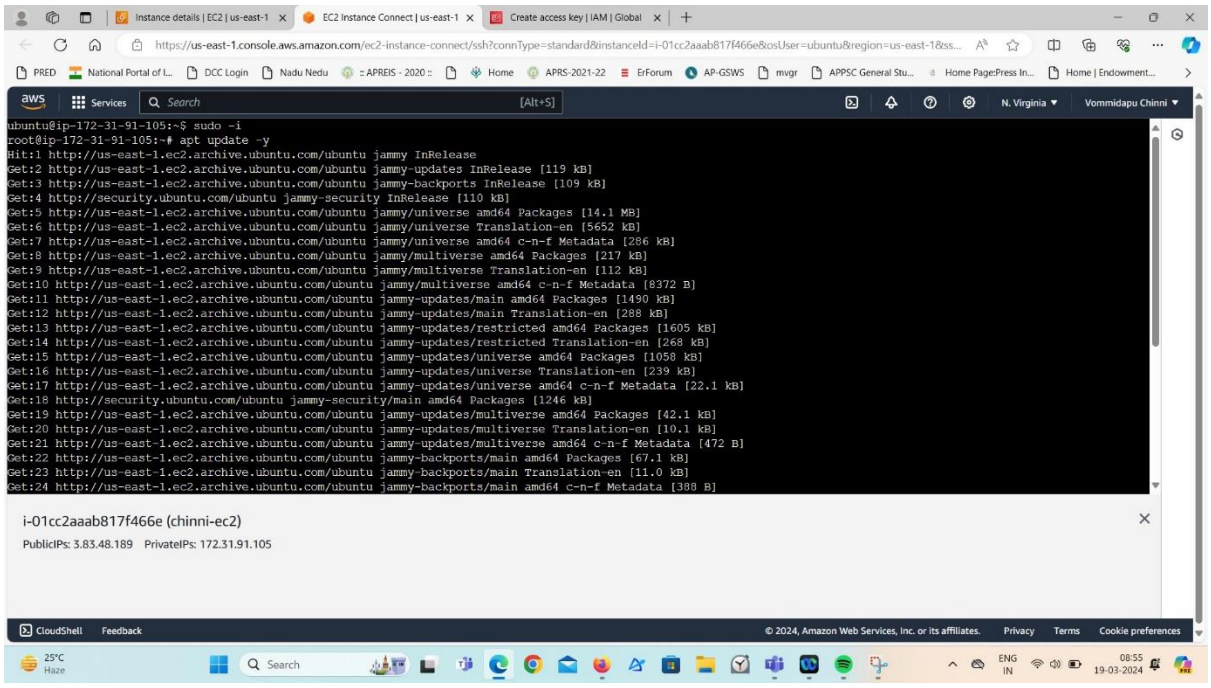


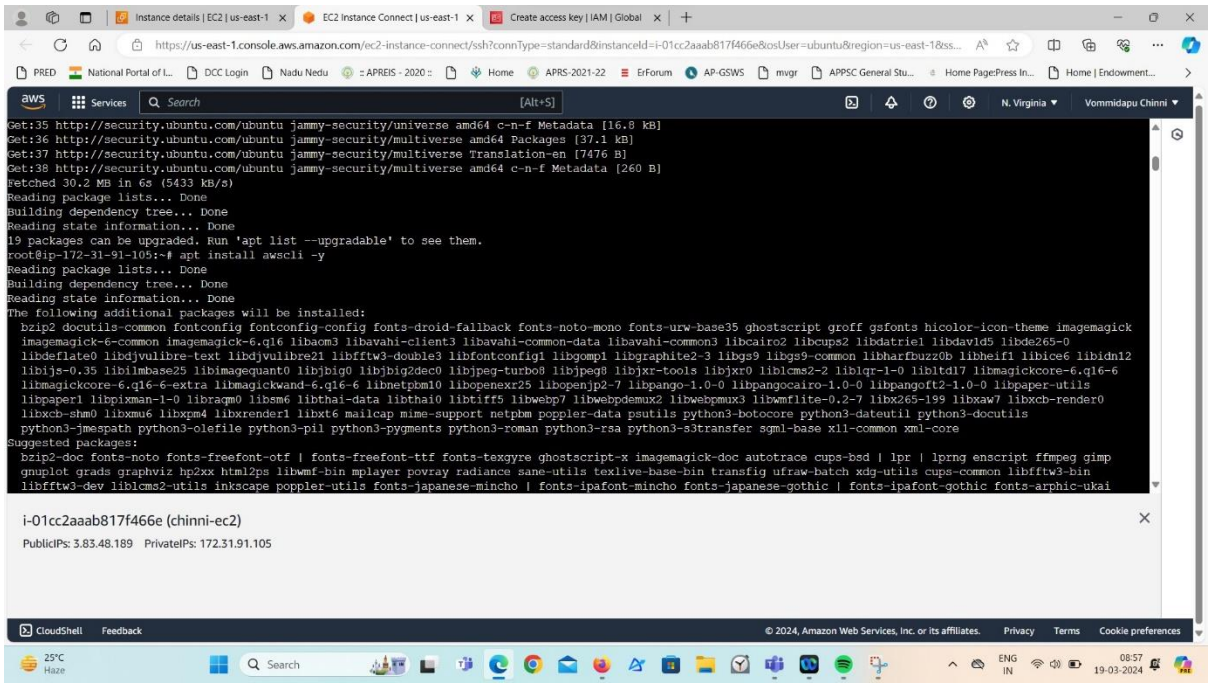


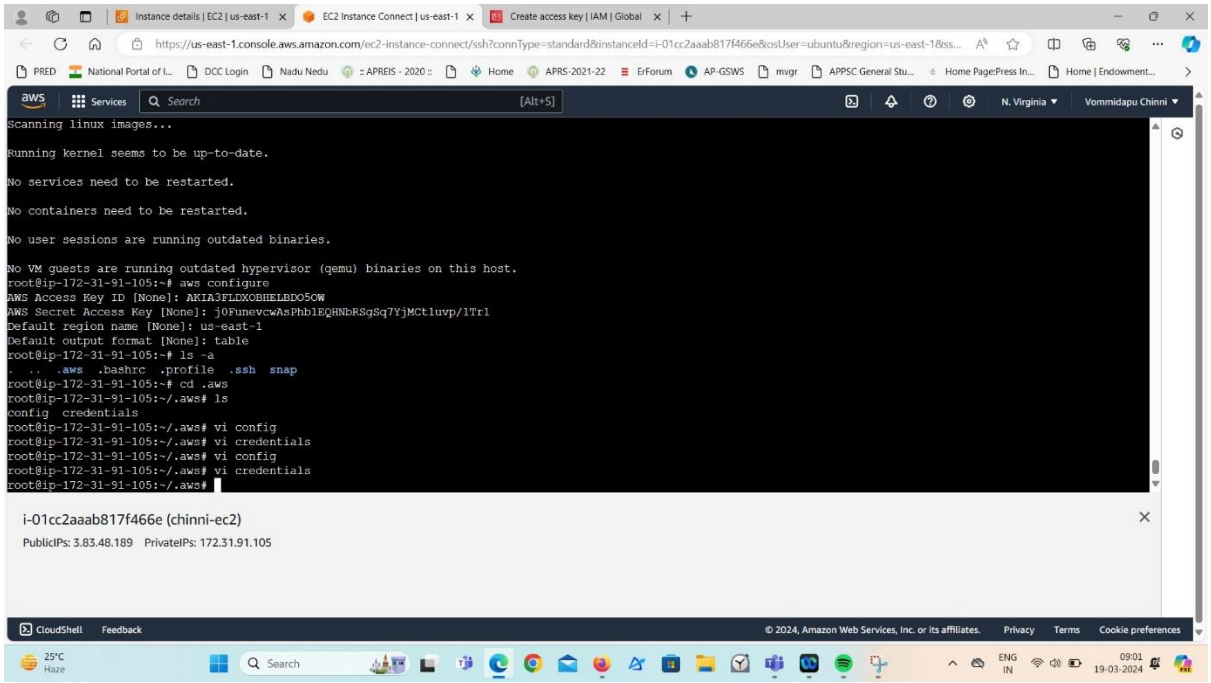
Step3: Connect to the server, Install awscli in server

a. Click on instance id, open it.
b. Click on connect, again click on connect.
c. Give commands like:
d. Sudo -i (convert normal user to root user).
e. apt update -y (to update).
f. apt install awscli(install awscli).

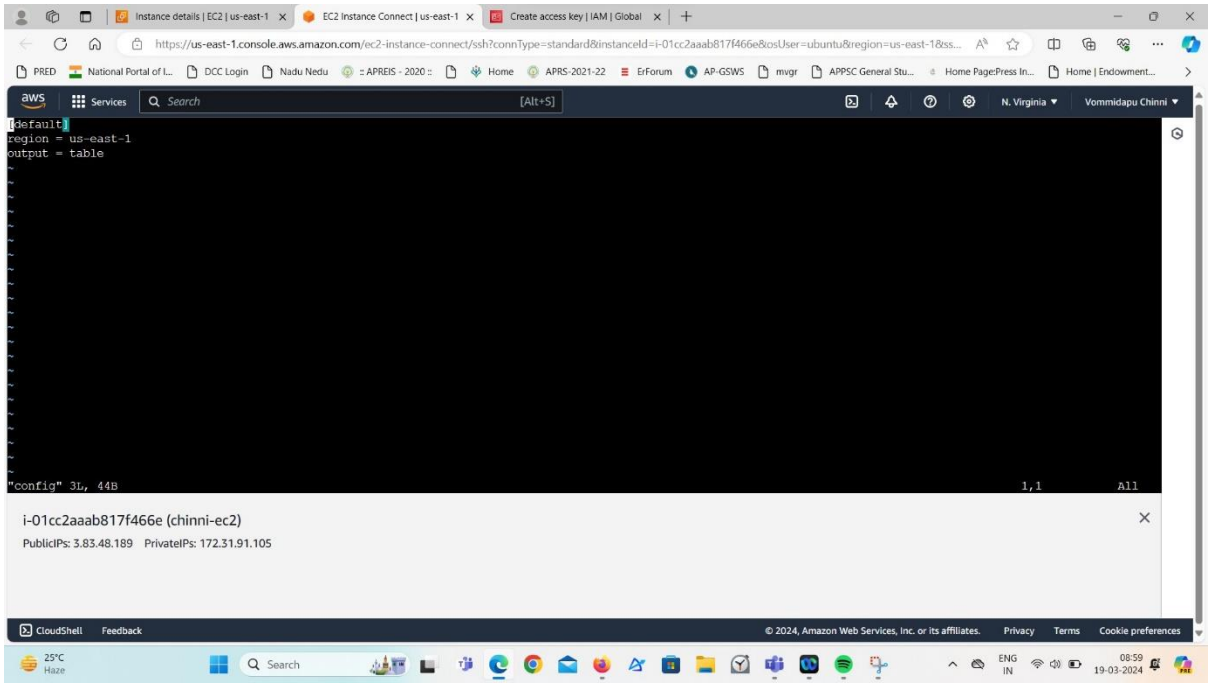
g. aws configure
h. give access key and secret access key.
i. Default region us-east-1
j. Format as table.
k. ls -a
l. cd .aws
m. ls



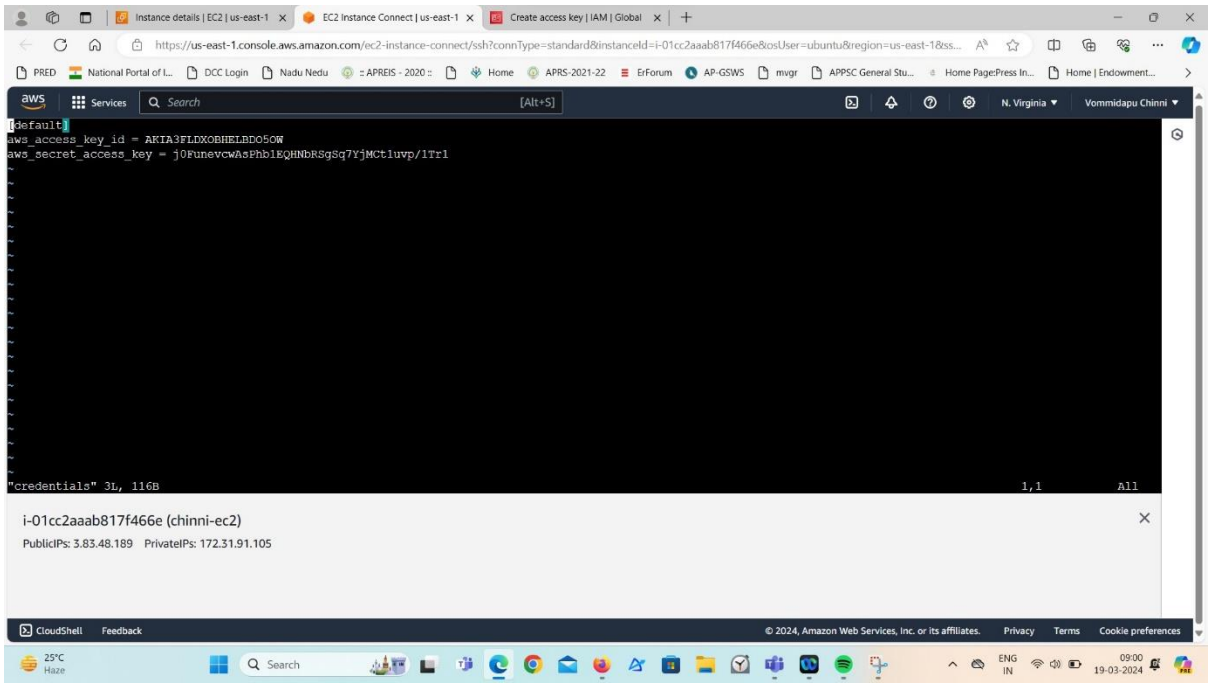



Instance details | EC2 | us-east-1
EC2 Instance Connect | us-east-1
Create access key | IAM | Global
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01cc2aaab817f466e&osUser=ubuntu&region=us-east-1&ss...
aws
Services
Search
[Alt+S]
N. Virginia
Vommidapu Chinni
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-91-105:~# aws configure
AWS Access Key ID [None]: AKIA3FLDXOBHELBDO5OW
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: table
root@ip-172-31-91-105:~# ls -a
. .. .aws .bashrc .profile .ssh snap
root@ip-172-31-91-105:~# cd .aws
root@ip-172-31-91-105:~/.aws# ls
config credentials
root@ip-172-31-91-105:~/.aws# vi config
root@ip-172-31-91-105:~/.aws# vi credentials
root@ip-172-31-91-105:~/.aws# vi config
root@ip-172-31-91-105:~/.aws# vi credentials
root@ip-172-31-91-105:~/.aws#
i-01cc2aaab817f466e (chinni-ec2)
PublicIPs: 3.83.48.189 PrivateIPs: 172.31.91.105
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C
Haze
Search
ENG
IN
09:01
19-03-2024

vi config:

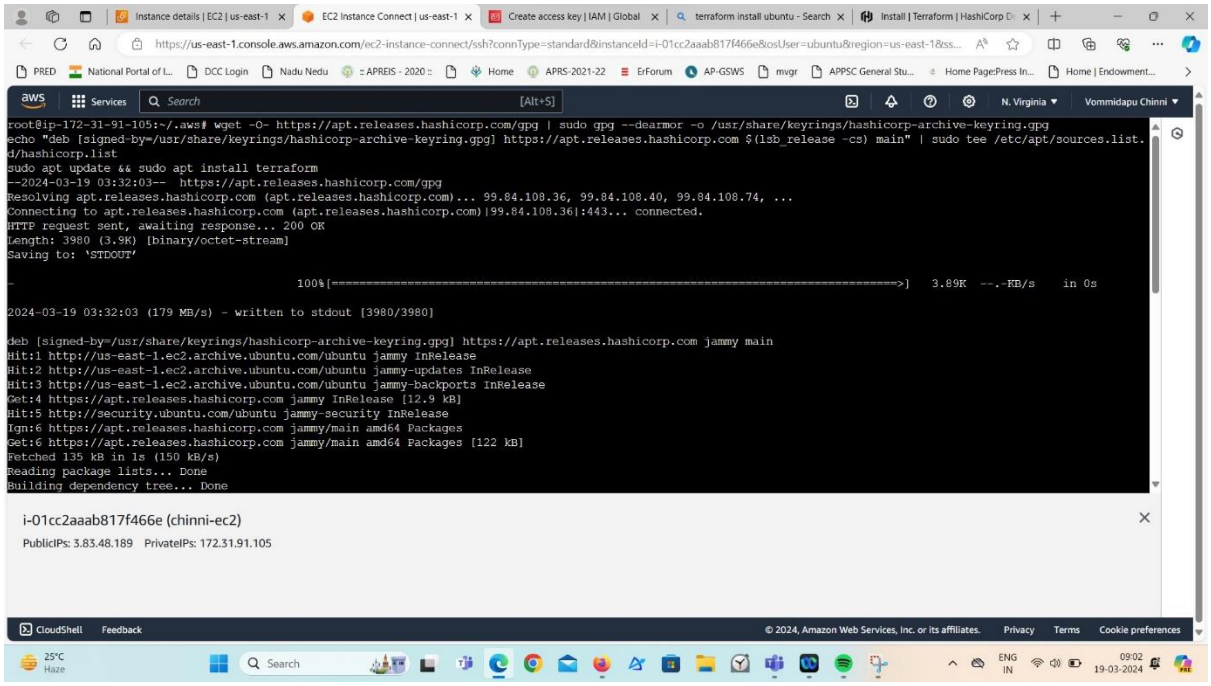

Instance details | EC2 | us-east-1
EC2 Instance Connect | us-east-1
Create access key | IAM | Global
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01cc2aaab817f466e&osUser=ubuntu&region=us-east-1&ss...
aws
Services
Search
[Alt+S]
N. Virginia
Vommidapu Chinni
[default]
region = us-east-1
output = table
"config" 3L, 44B
1,1
All
i-01cc2aaab817f466e (chinni-ec2)
PublicIPs: 3.83.48.189 PrivateIPs: 172.31.91.105
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C
Haze
Search
ENG
IN
08:59
19-03-2024

vi credentials:

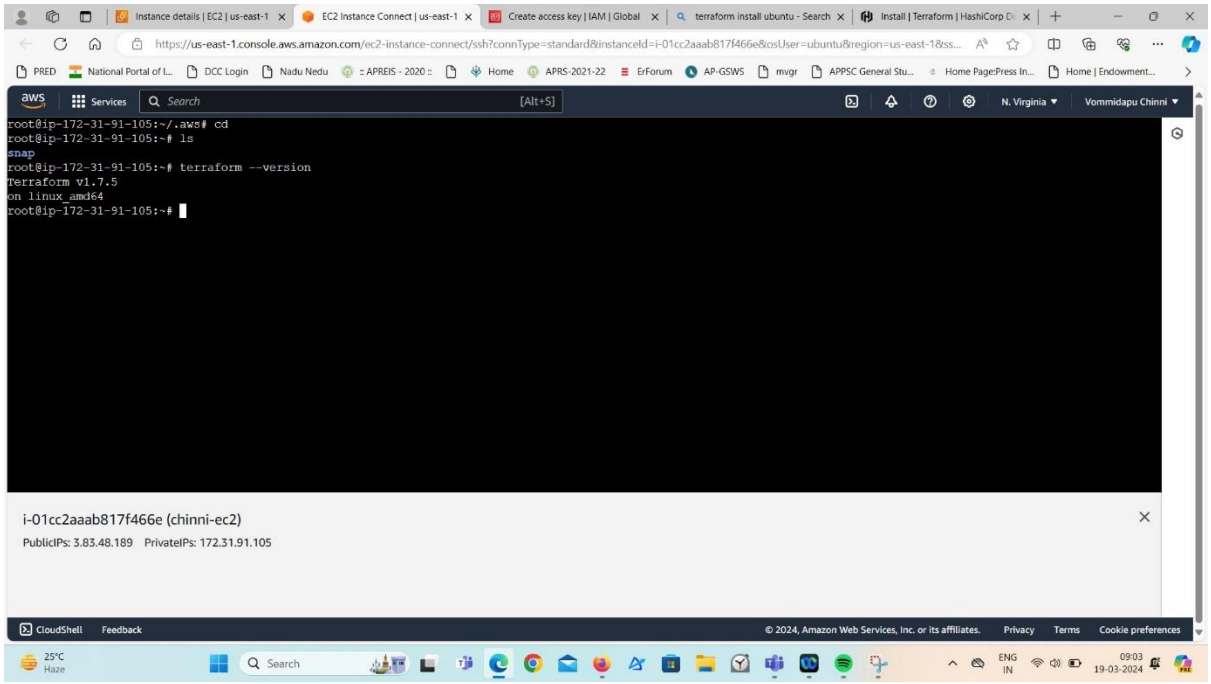


Step4: Install terraform in server

- By using : wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
- echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
- sudo apt update && sudo apt install terraform



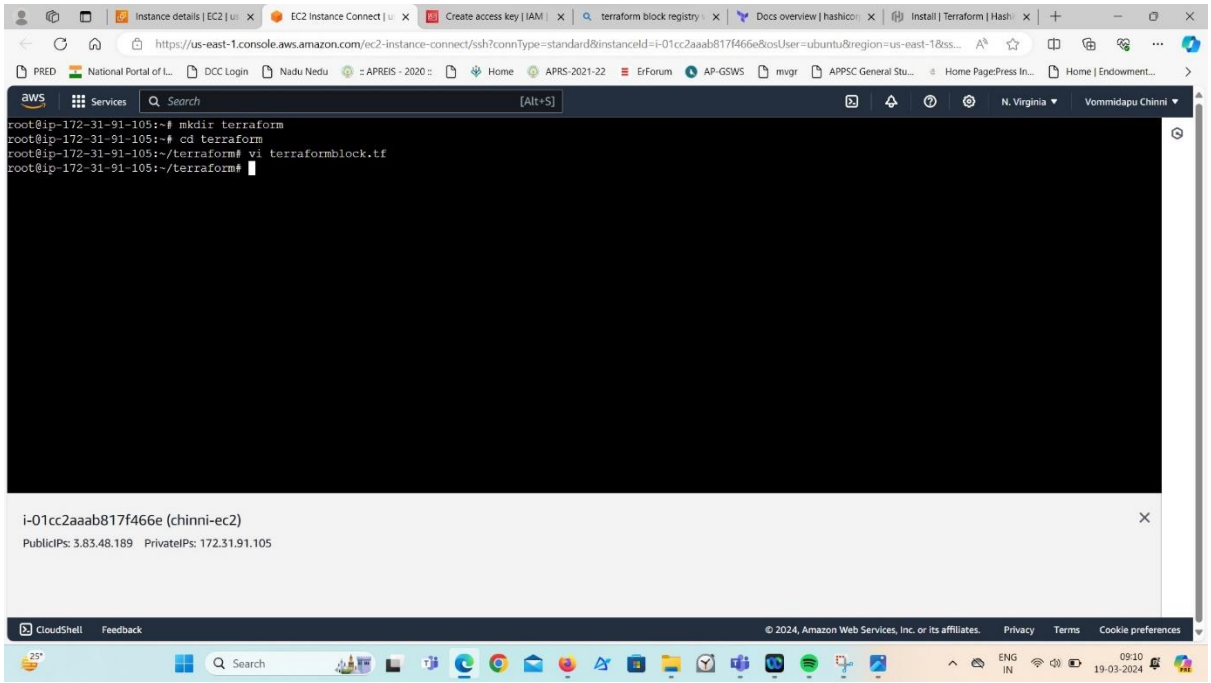
- cd (use to change directory)
- terraform –version (to know terraform version).

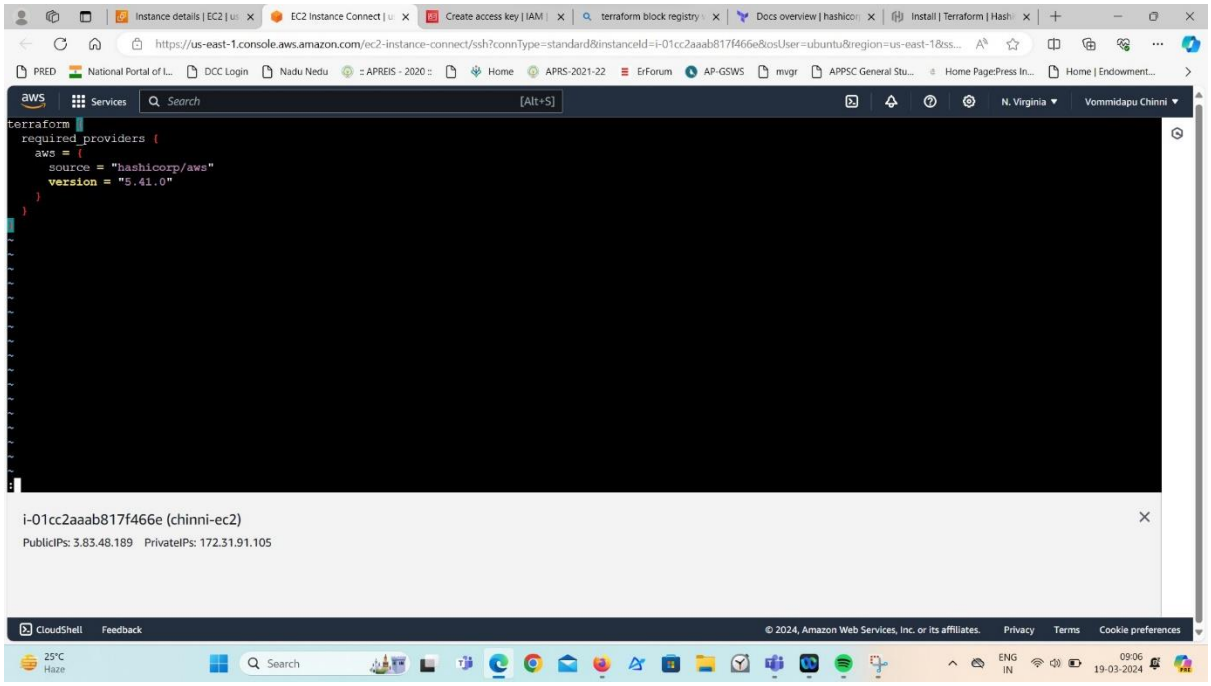


Step5: Create terraform directory:

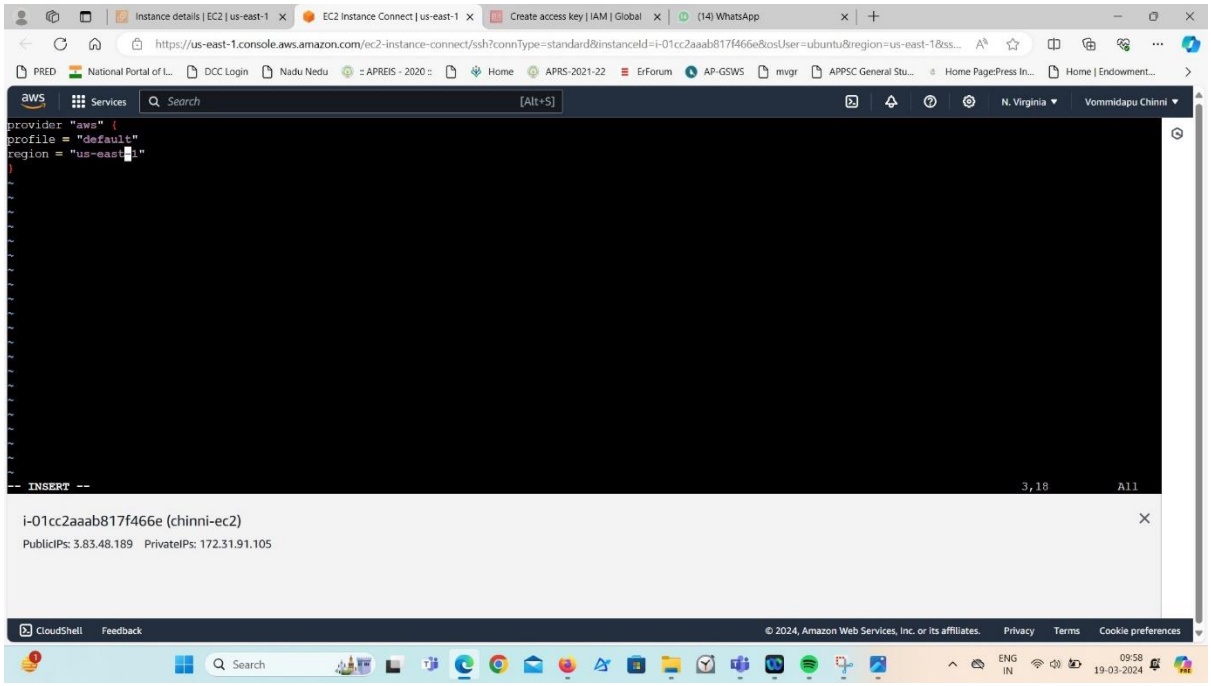
- mkdir terraform
- cd terraform

➢ Create terraform block using vi terraformblock.tf



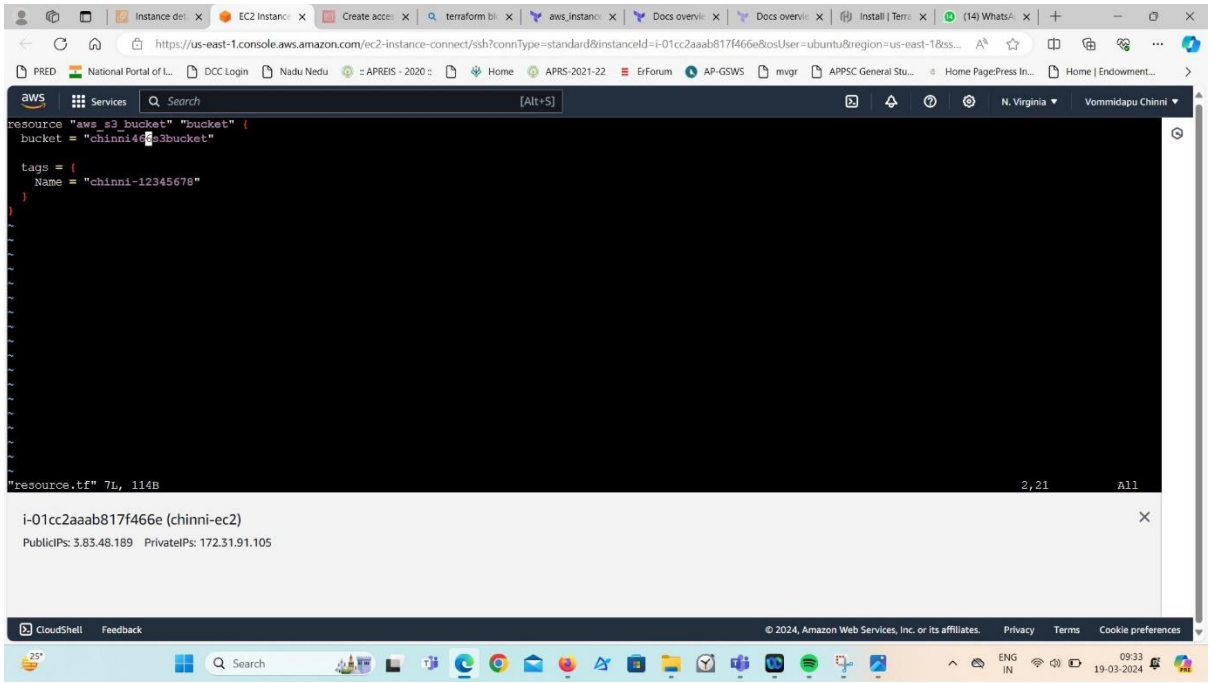


- Create provider block using vi provider.tf

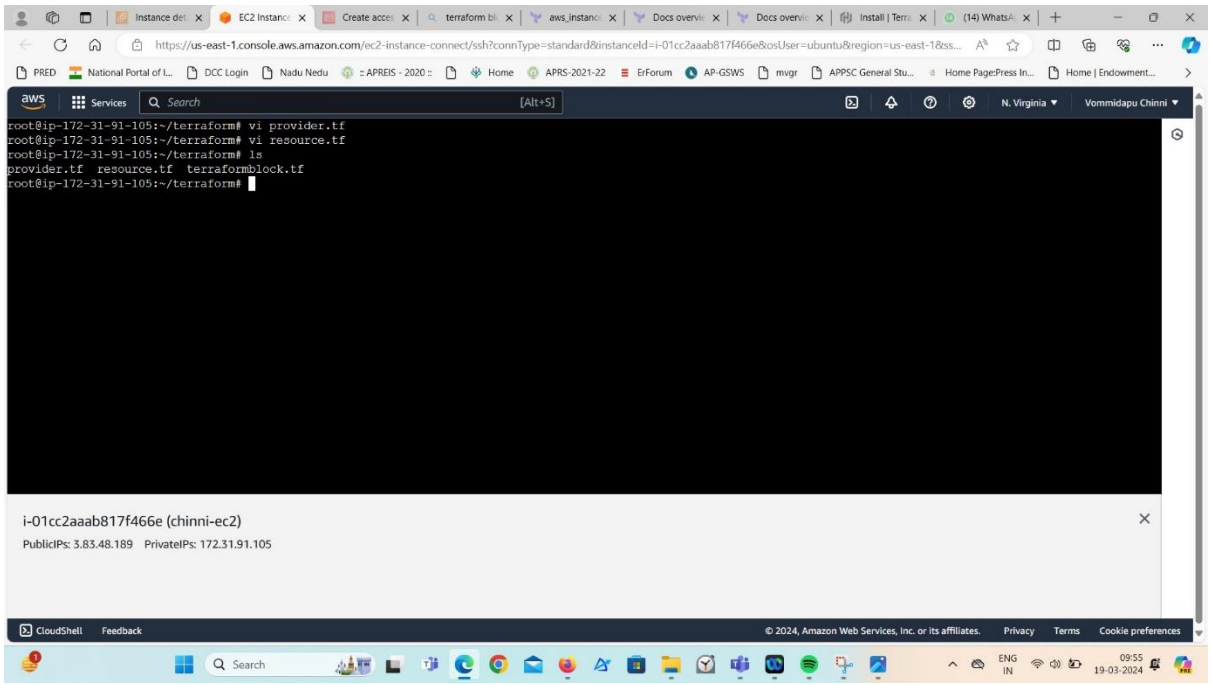


- Create resource block using vi resource.tf

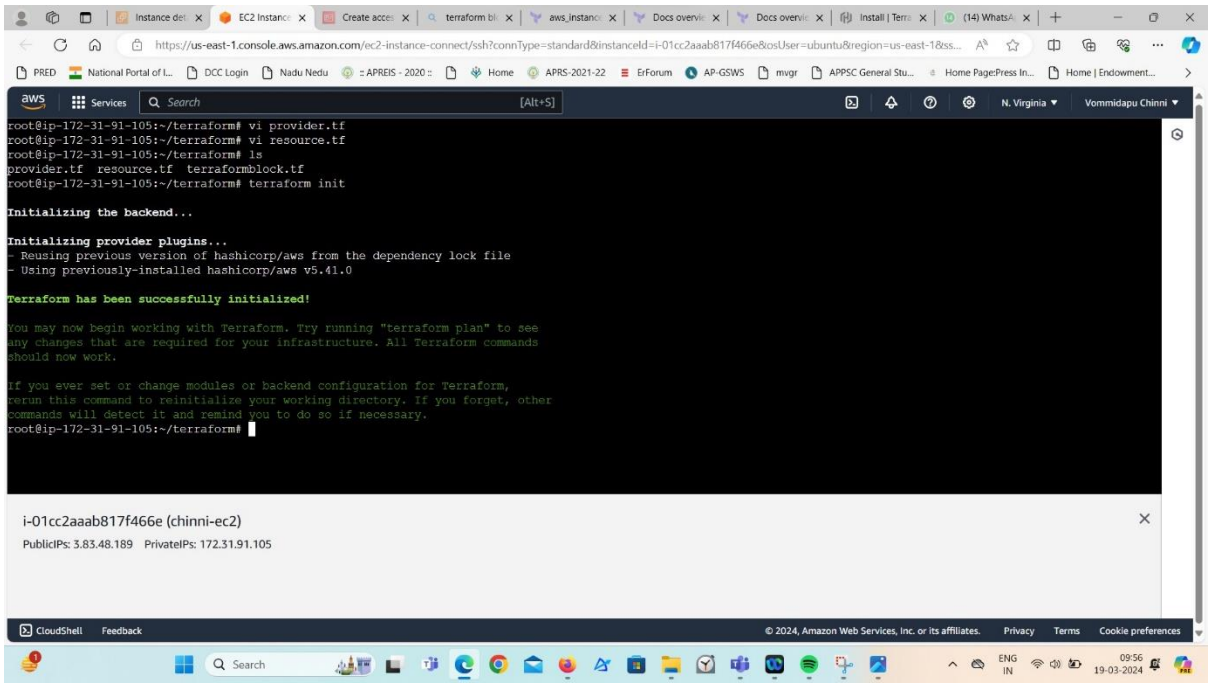
    - For create an S3 bucket we take snippet from terraform registry.
    - Creating a bucket using a name chinni46s3bucket.



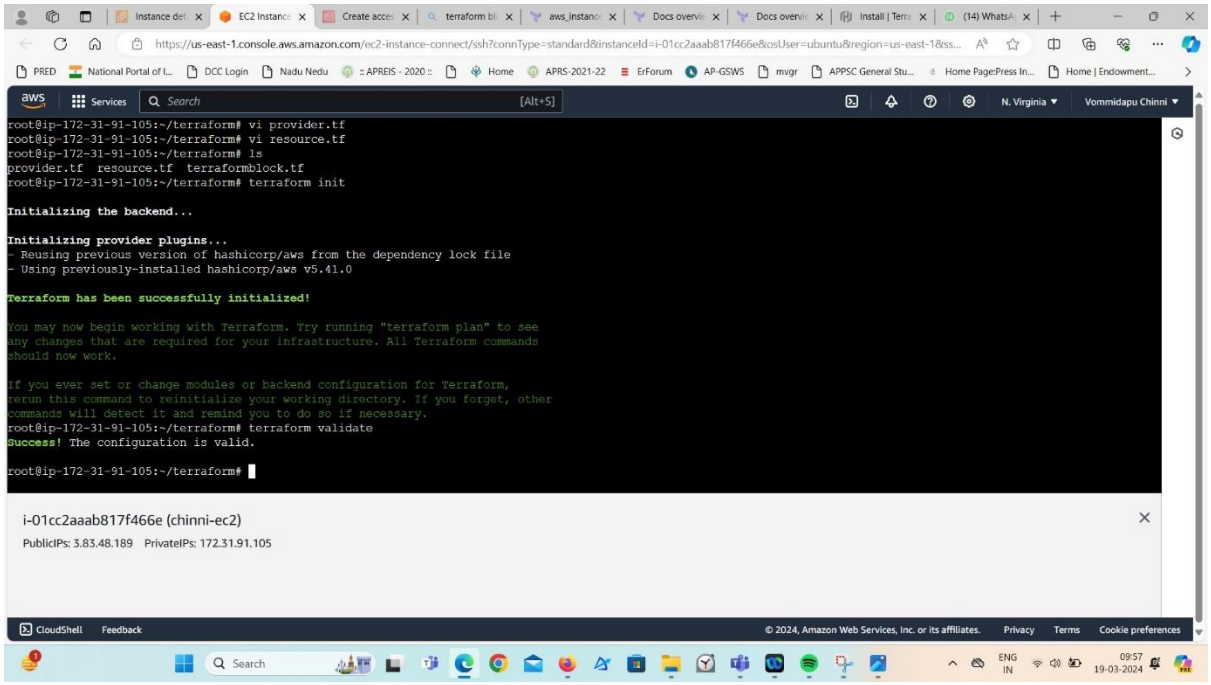
- Up to now we created three blocks, by using ls we can list the files in terraform directory.



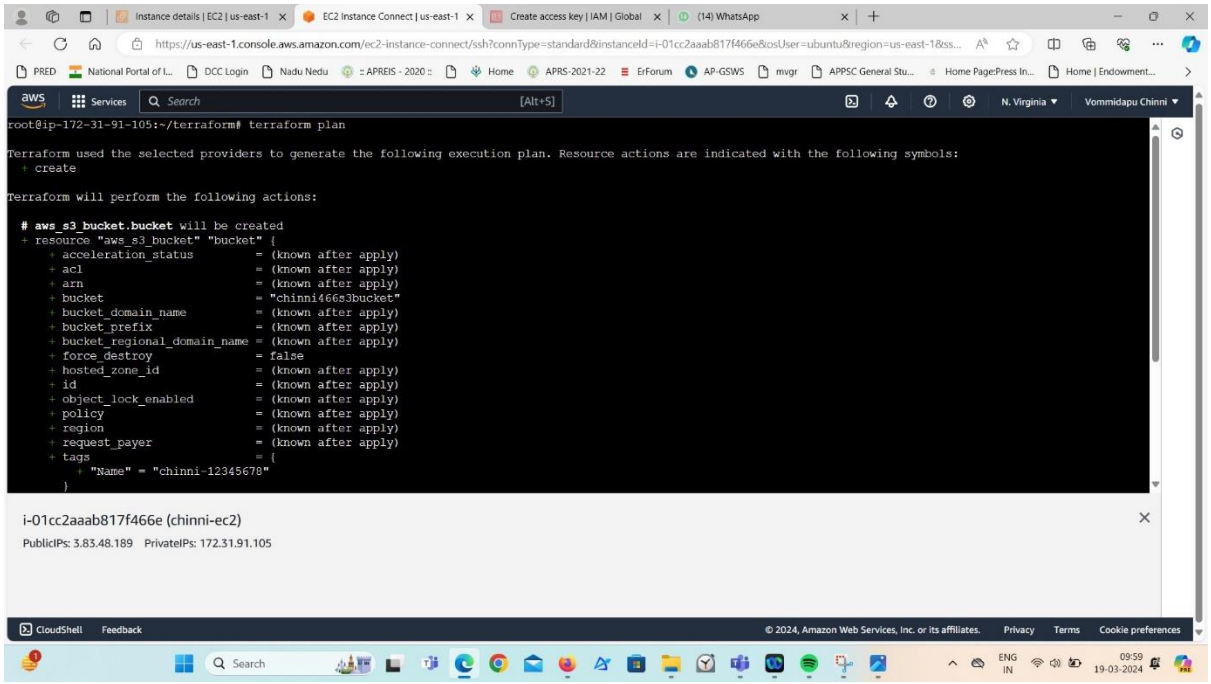
- terraform init (Preparing our working directory for other commands).



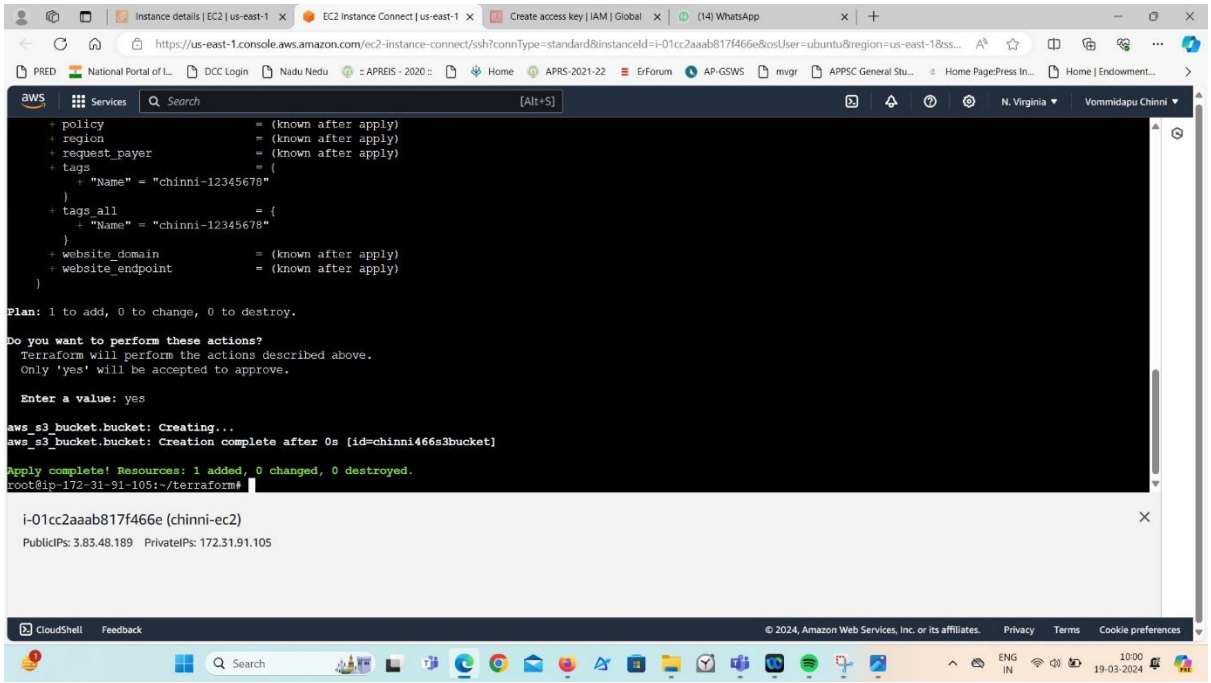
- terraform validate (Check whether the configuration is valid or not).



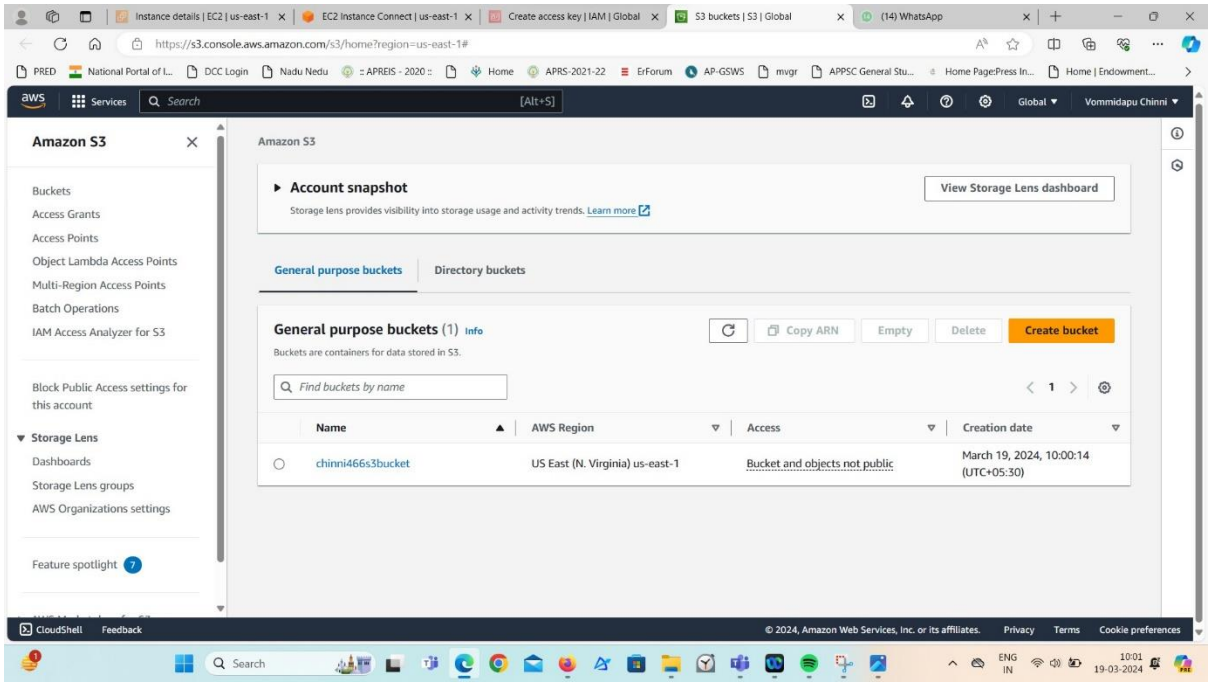
- terraform plan (show changes required by the current configuration).



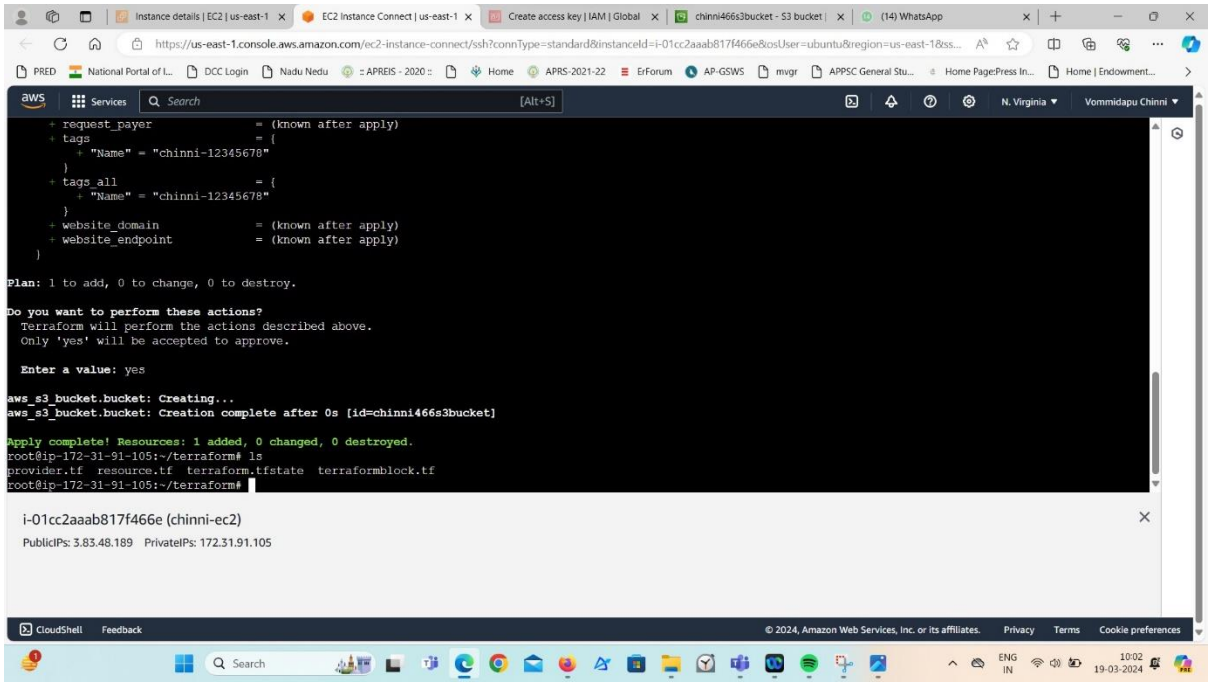
- terraform apply (create the infrastructure).



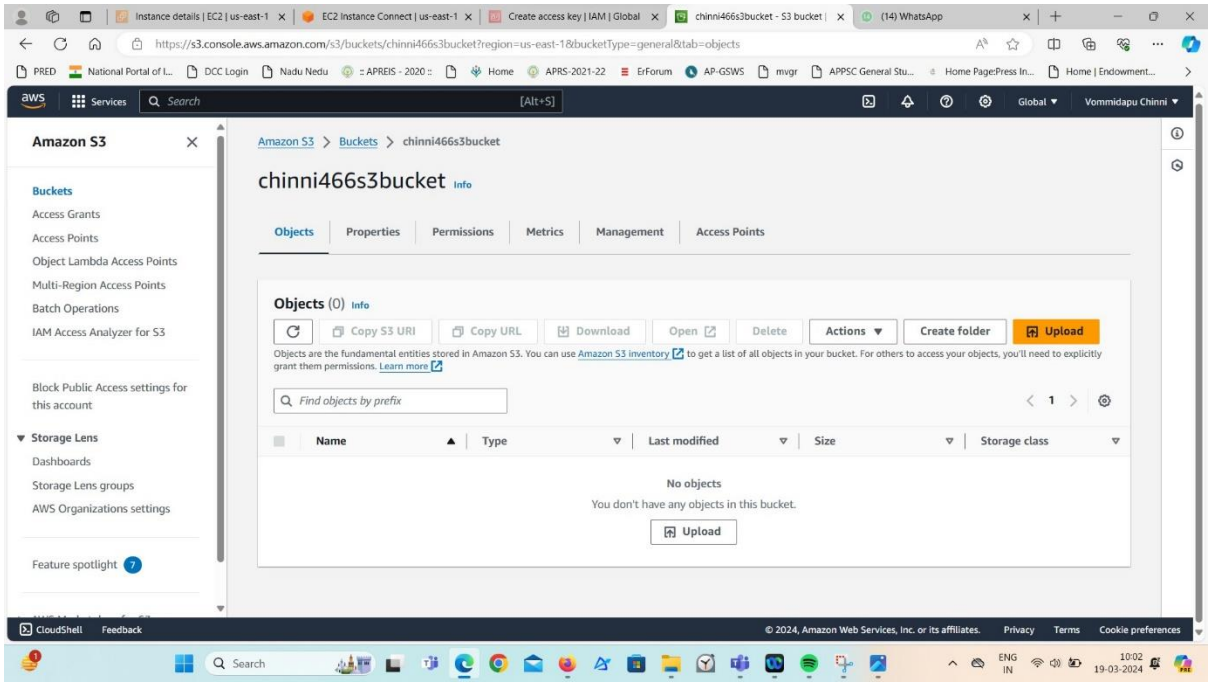
- Open S3 dashboard.
- Click on buckets.
- We can see that bucket is created with a name chinni466s3bucket.



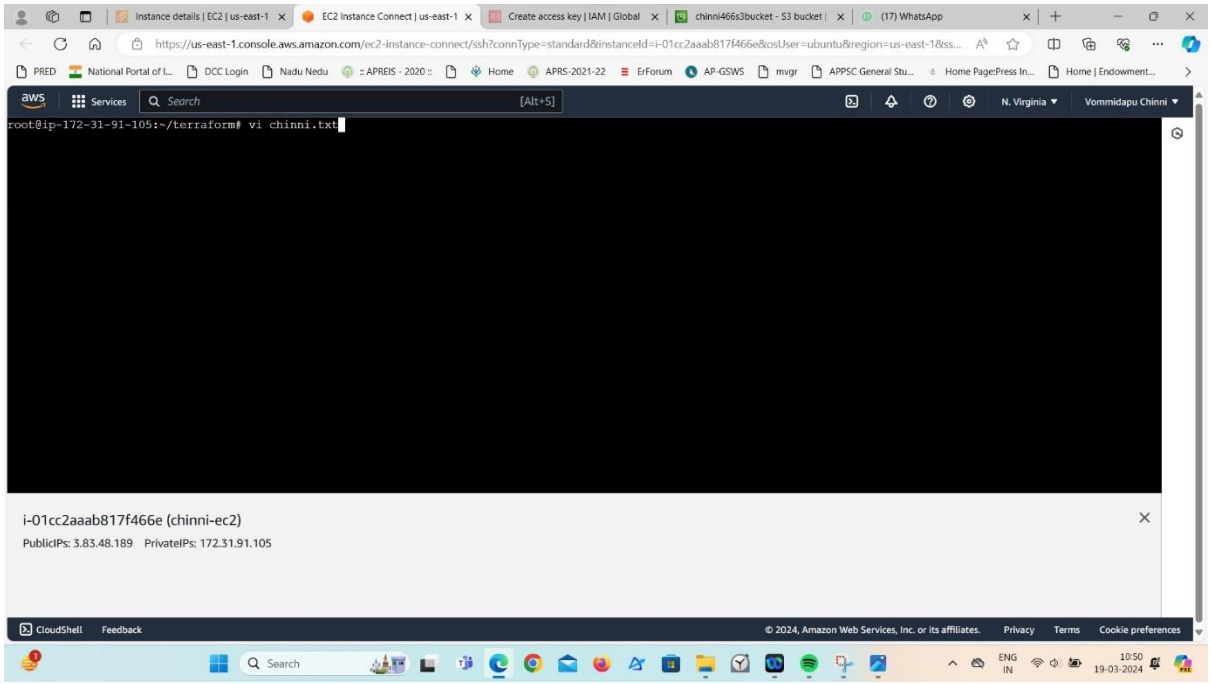
- Give command like ls, we can see four files, that terraform.tfstate file contains that what was created in it.



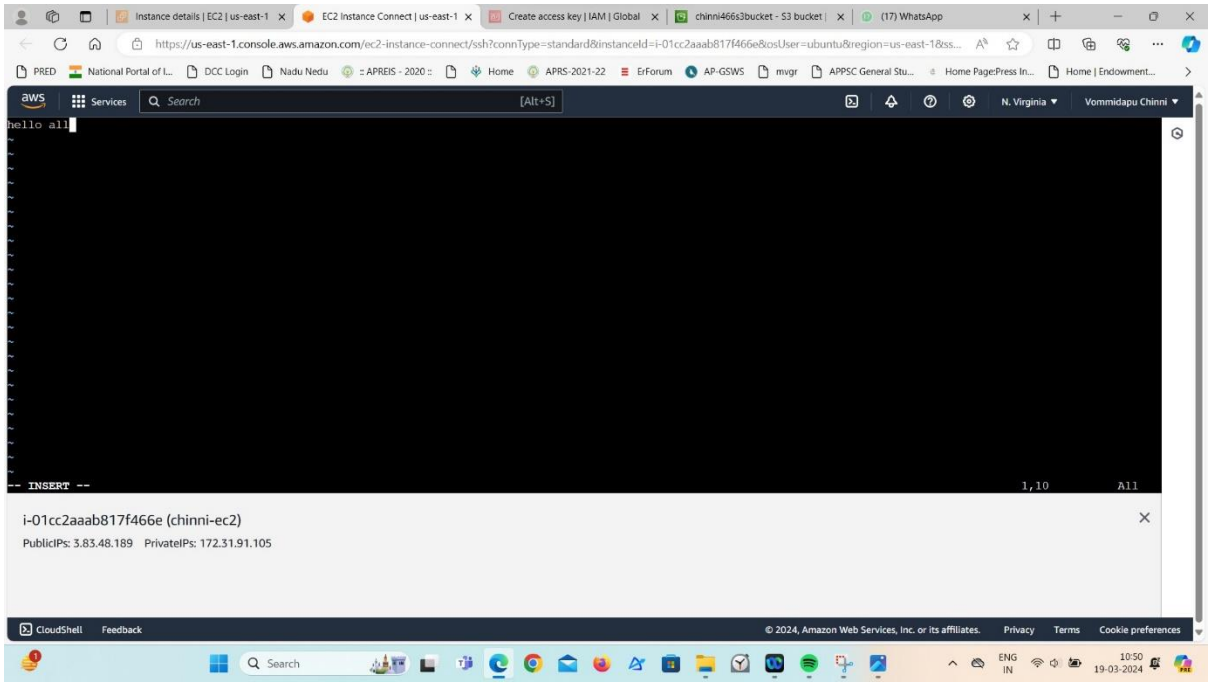
- Upload an object/file to the S3 bucket (chinni466s3bucket).
    - Before that open chinni466s3bucket, there is nothing object/file in that bucket.



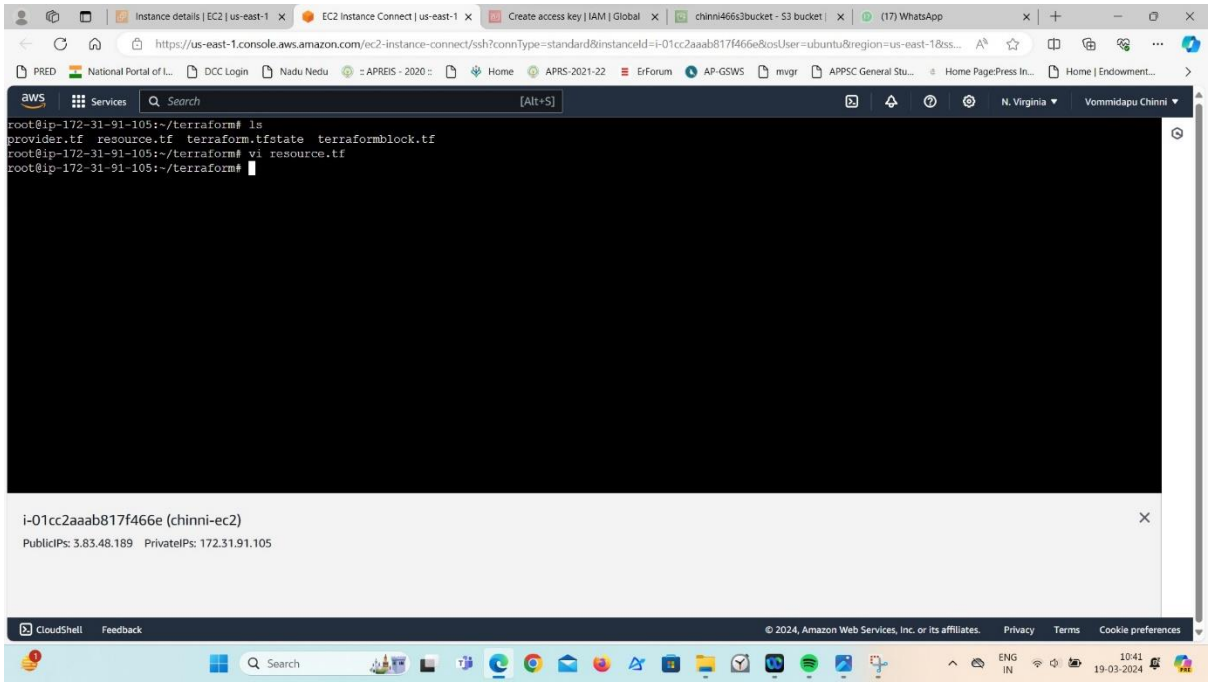
- Create a text file in that directory, by giving the name chinni.txt (vi chinni.txt).



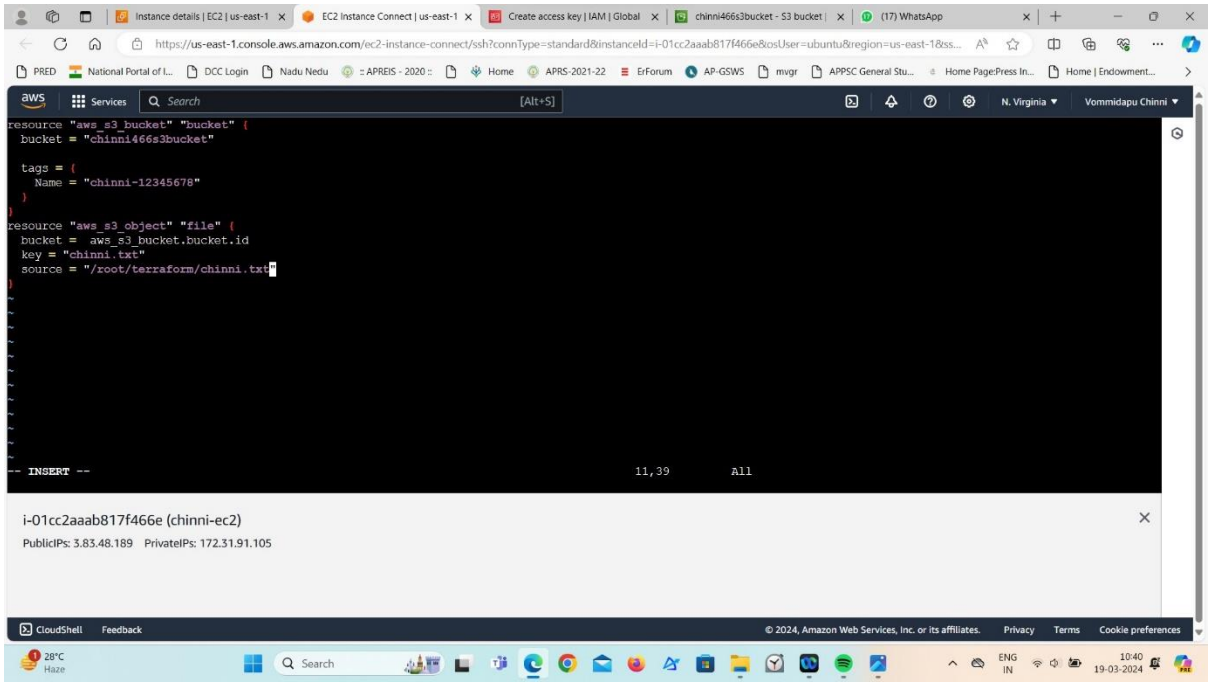
- In that enter the text (hello all).



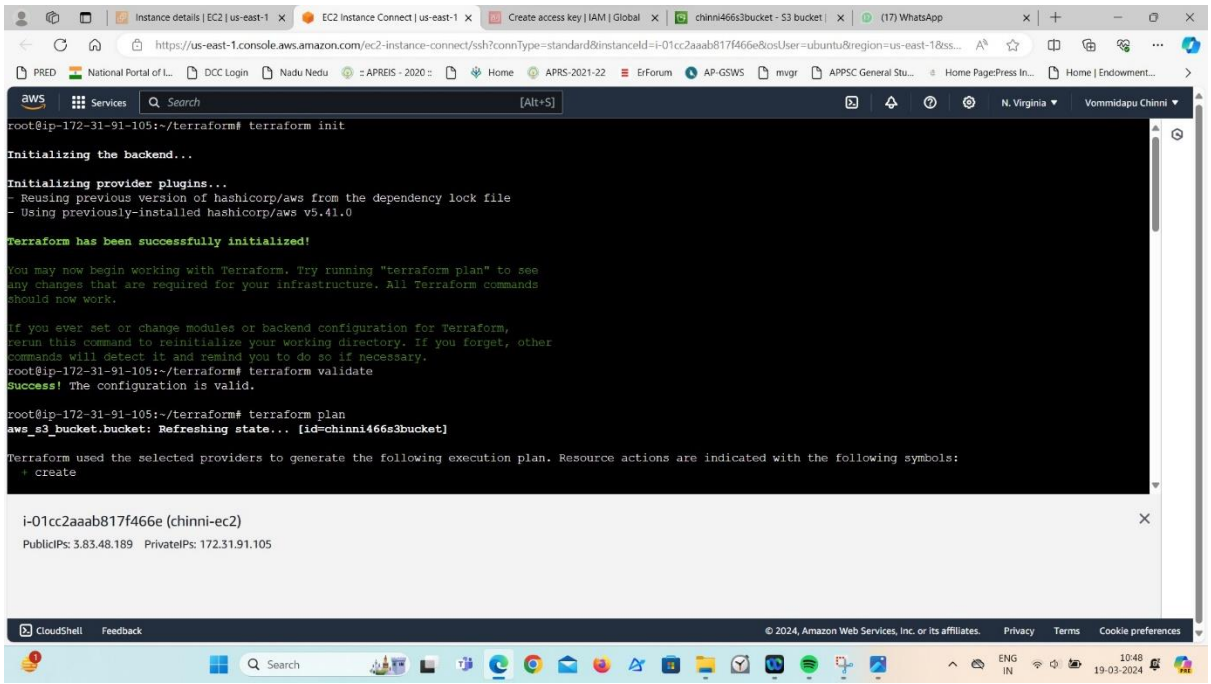
- Open vi resource.tf



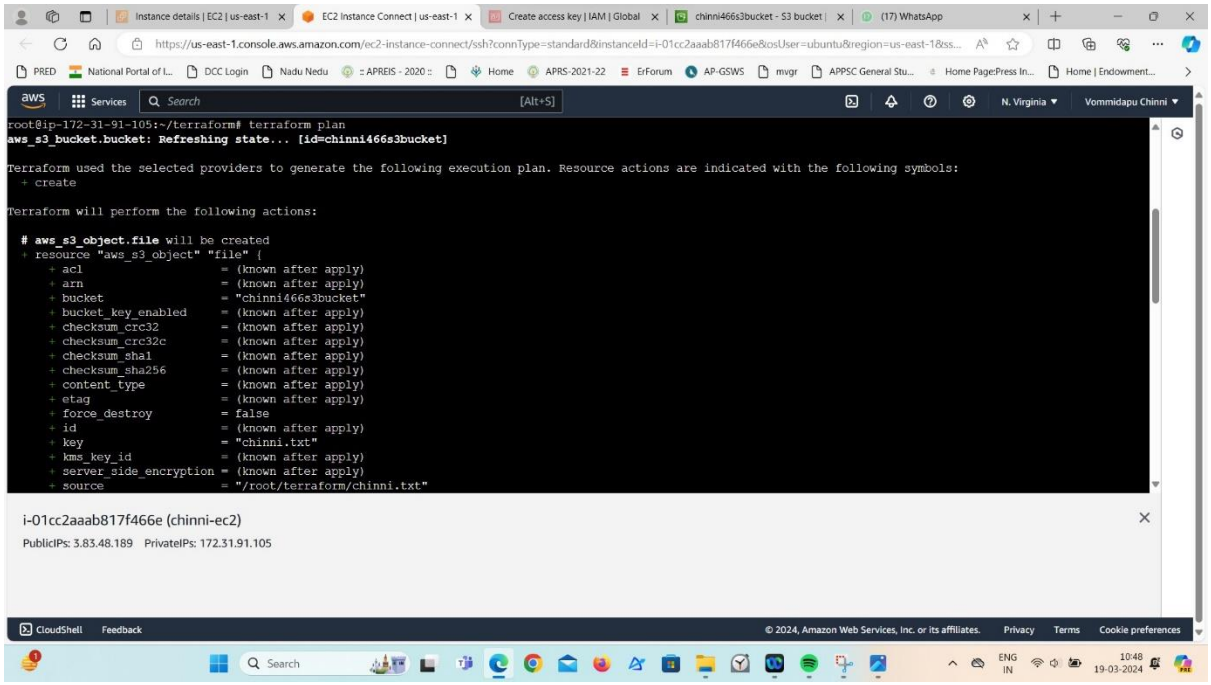
- Open terraform registry copy the snippet for S3 bucket object upload.



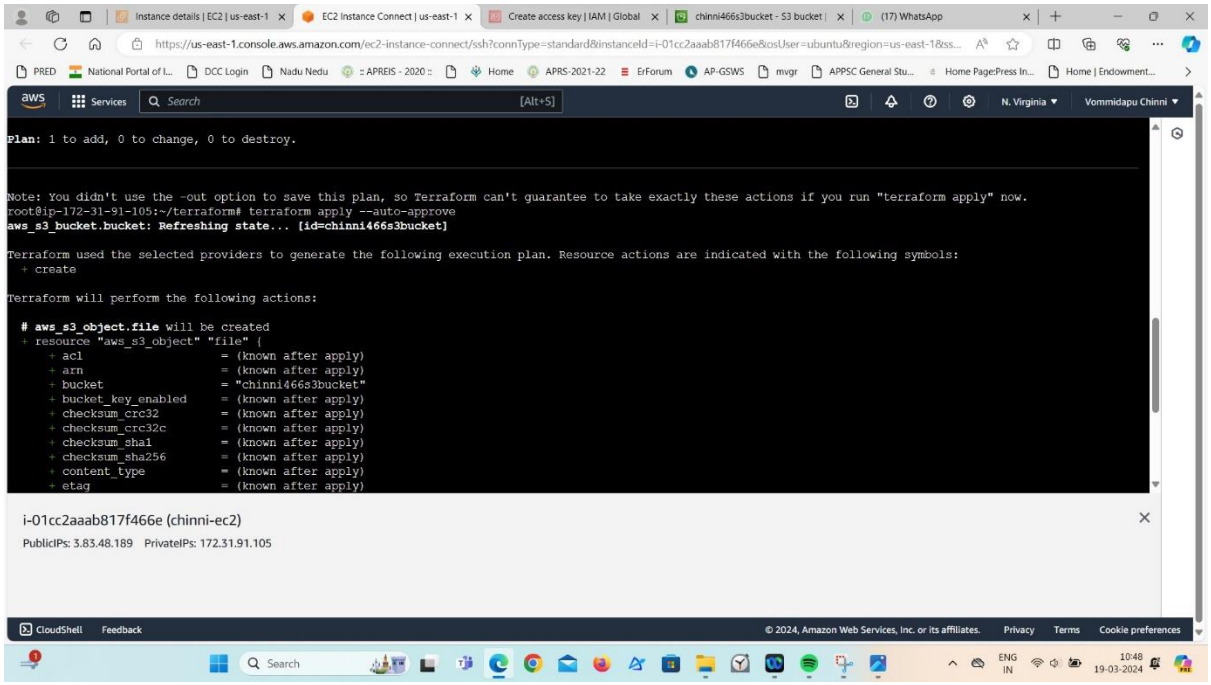
- terraform init and terraform validate.

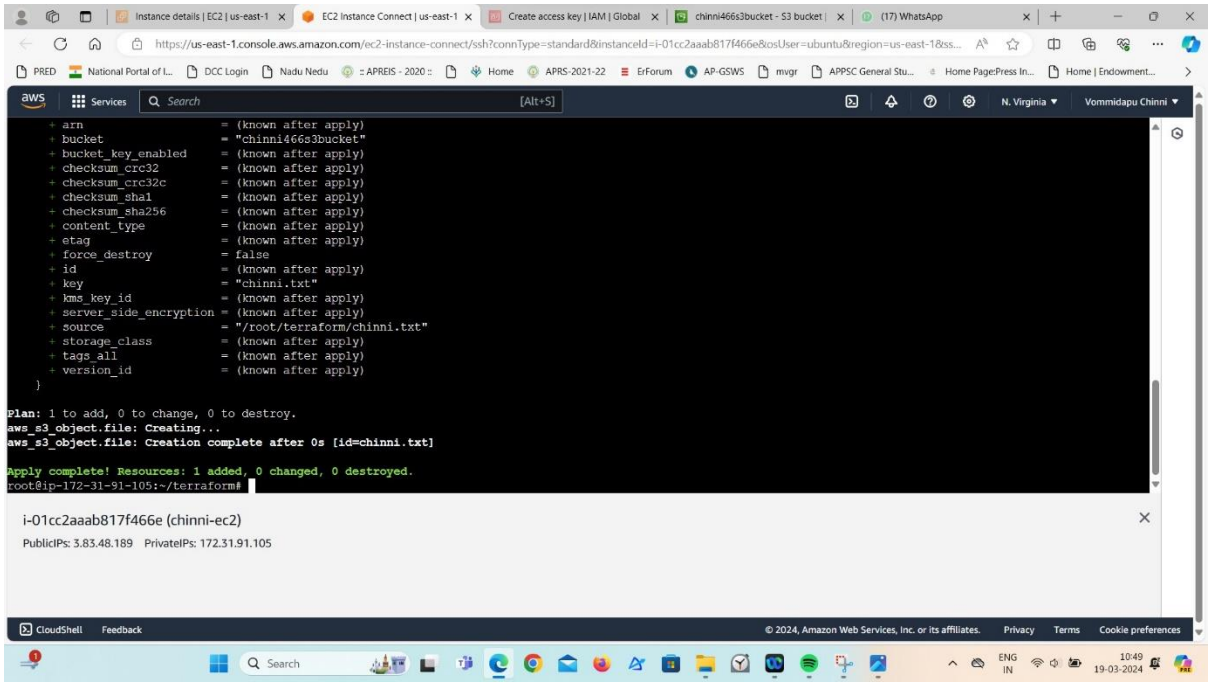


- terraform plan

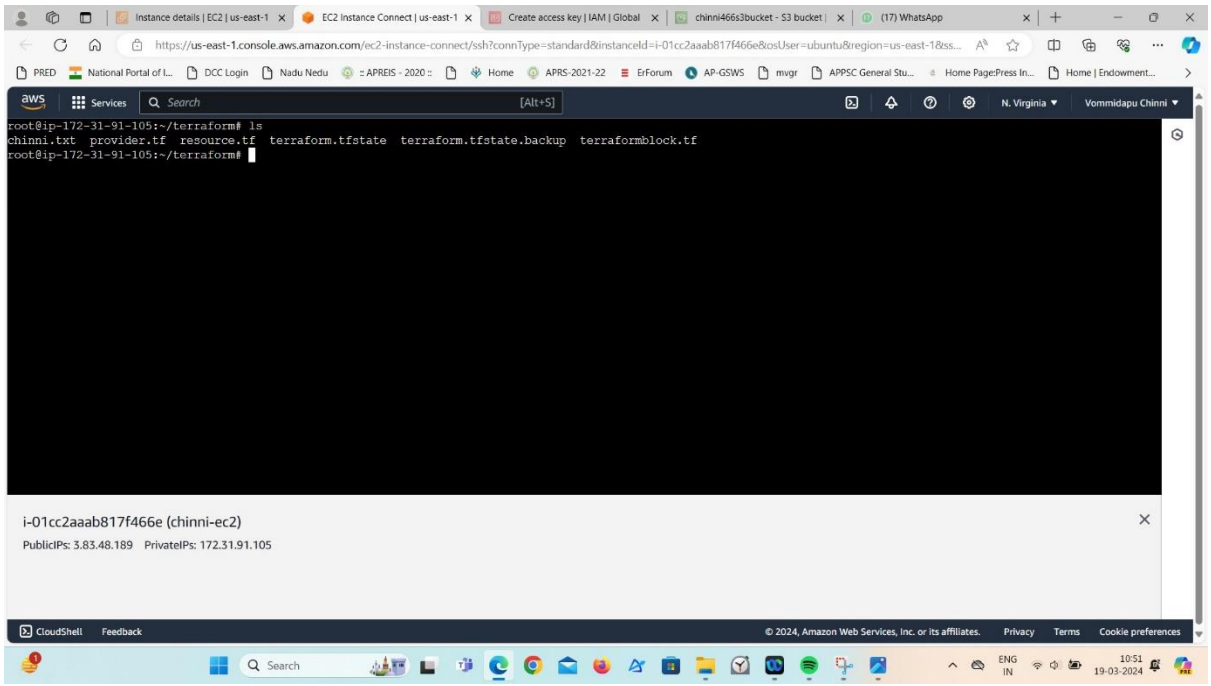


- terraform apply –auto-approve (without asking yes it can approve it).

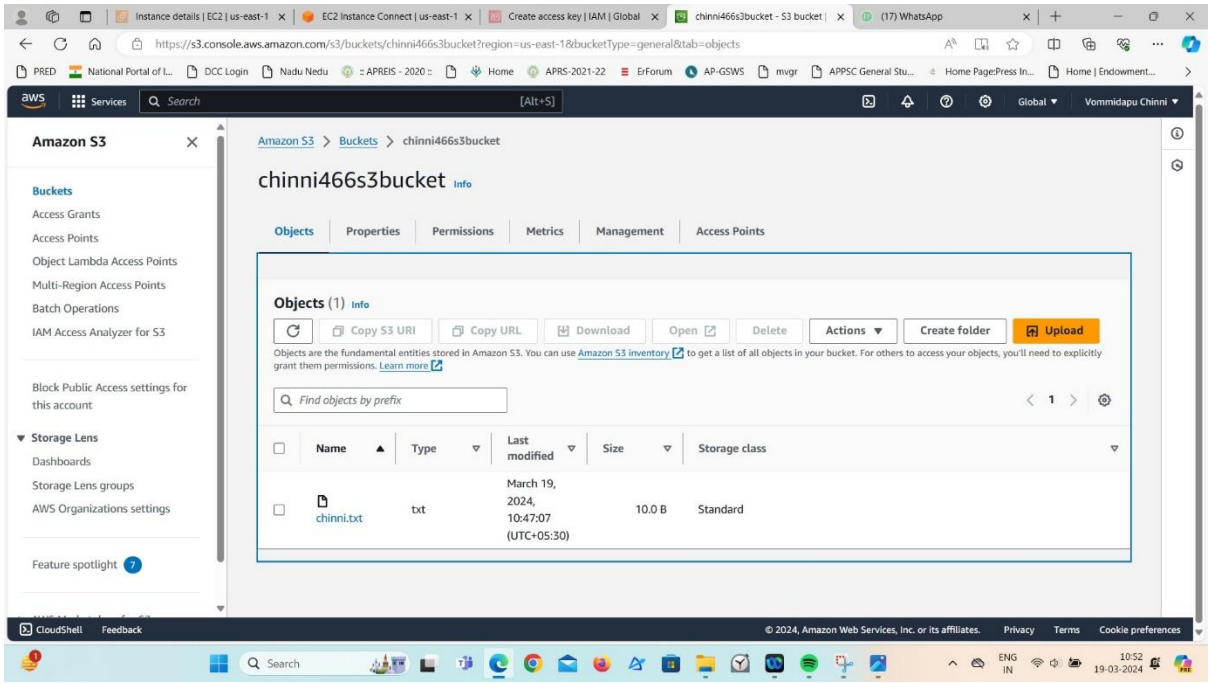




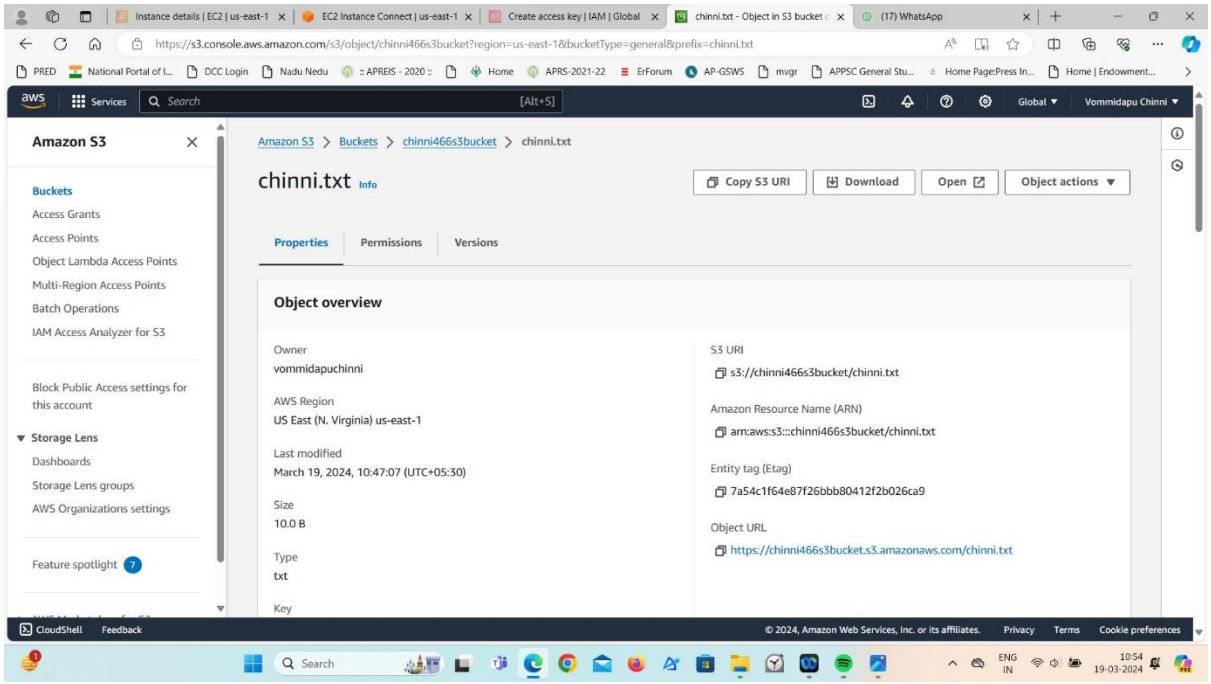
- Up to know what we created we can see using ls command



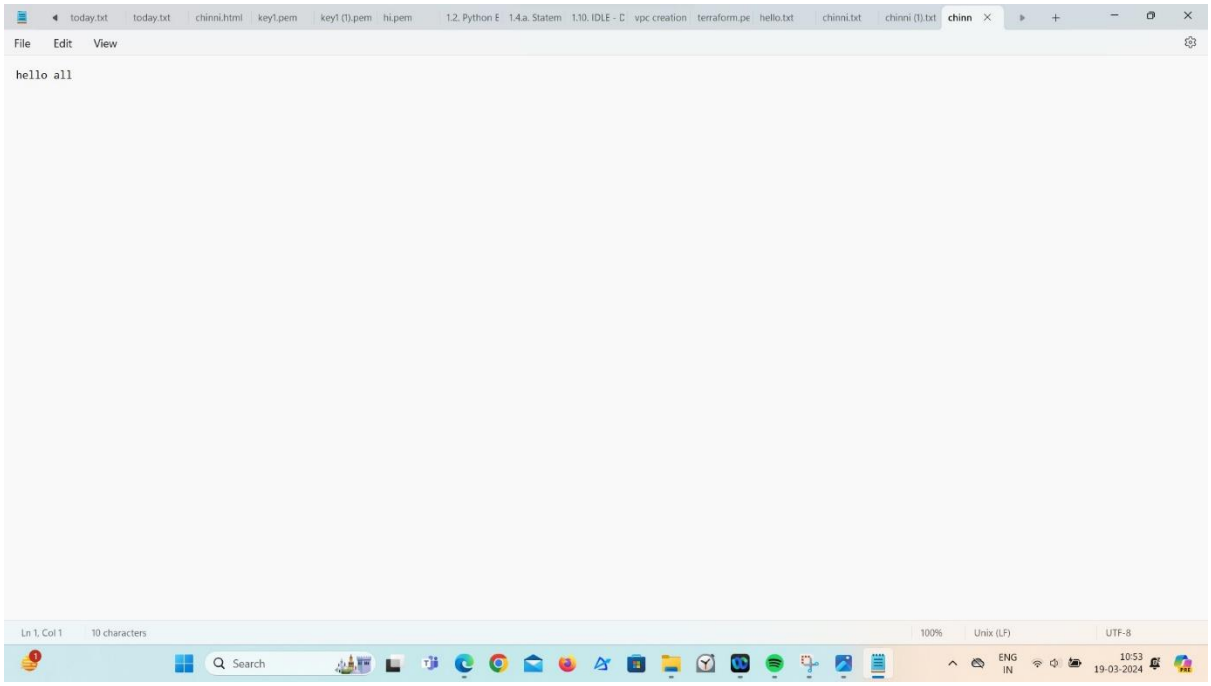
- Open S3 dashboard, open chinni466s3bucket, we can see a object/file uploaded in it. What was created in the directory (chinni.txt).



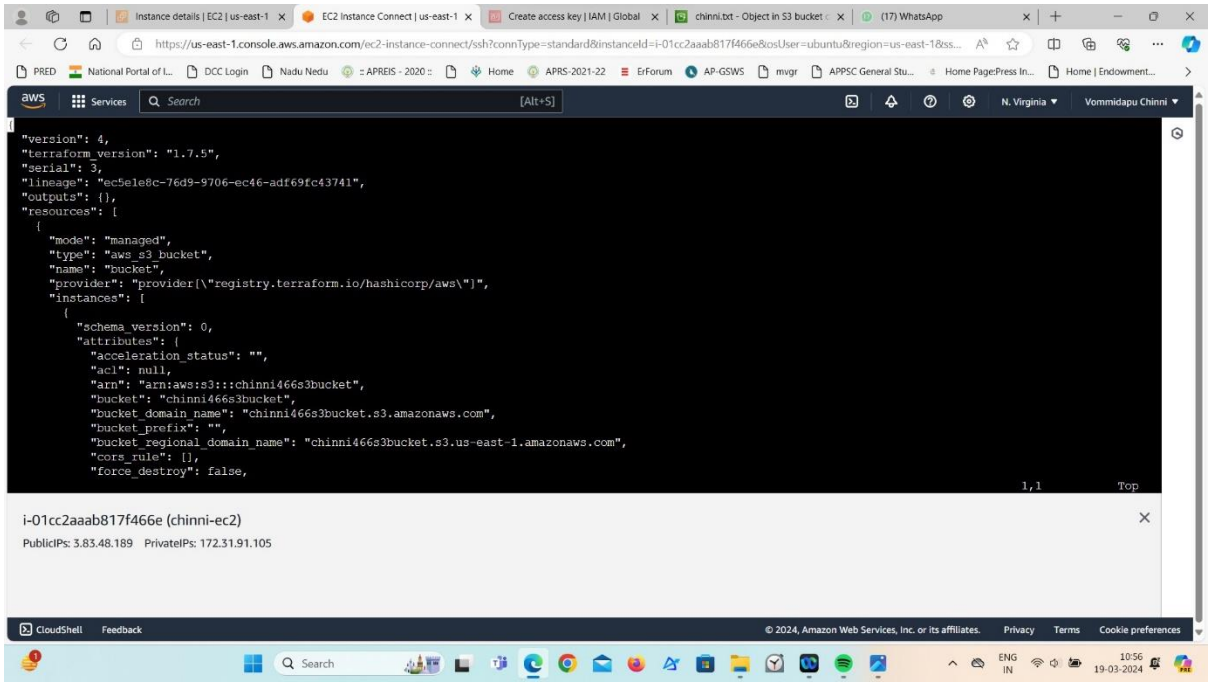
- Open the file chinni.txt



- Click on open which is on the right side of the corner.
- We can see that chinni.txt is downloaded, open it, we can see that what is there in text file.



- We can open the terraform.tfstate file by using the command vi terraform.tfstate.



At last we have destroy by using the command terraform destroy (destroy previously created infrastructure).